ESD-TR-87-263

Annual Report

Knowledge-Based System Analysis and Control Defense Switched Network Task Areas

30 September 1987

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

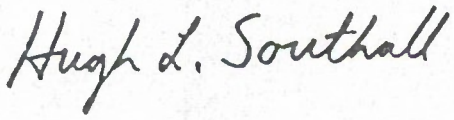
LEXINGTON, MASSACHUSETTS

Prepared for the Department of the Air Force
under Electronic Systems Division Contract F19628-85-C-0002.

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, with the support of the Department of the Air Force under Contract F19628-85-C-0002.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER



Hugh L. Southall, Lt. Col., USAF
Chief, ESD Lincoln Laboratory Project Office

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

KNOWLEDGE-BASED SYSTEM ANALYSIS AND CONTROL DEFENSE SWITCHED NETWORK TASK AREAS

ANNUAL REPORT SUBMITTED TO
WILLIAM COHEN
DCEC R610
1860 WIEHLE AVENUE
RESTON, VA 22090-5500

H.M. HEGGESTAD
Group 21

1 OCTOBER 1986 — 30 SEPTEMBER 1987

ISSUED 24 FEBRUARY 1988

LEXINGTON MASSACHUSETTS

ABSTRACT

Extensive modifications have been made in the existing Defense Switched Network (DSN) Call-by-Call Simulator (CCSIM) to permit its use in learning how to apply Network Management techniques in the DSN. The initial capabilities and purposes of the CCSIM are described, and the requirements for the modifications are explained. In particular, the current DSN Network Management control options are defined and explained. The upgrading of CCSIM has been completed, and some hundreds of simulation runs have been carried out and analyzed to date. In support of this analysis, it has been necessary to augment the traditional Grade of Service (GOS) measure of network performance with a pair of new measures expressing actual user experience with the net, namely Call Failure Rate (CFR) and Mean Tries for Success (MTFS). A set of basic recommendations has been derived for near-term manual DSN Network Management actions under certain failure scenarios. Concurrently, a prototype has been developed for a knowledge-based Expert System to incorporate and organize the information and techniques being gathered, so that this knowledge can be made available for reference and advice for DSN Network Managers in the field.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Background

The DCEC-sponsored Knowledge-Based Systems Analysis and Control Program at Lincoln Laboratory was established to address an important problem area associated with implementing the Defense Switched Network, namely learning how to apply network management techniques to correct problems and avoid disasters. Installation of modern computer-controlled DSN switches is already in progress, and the new network is much more complex than the AUTOVON system it is replacing, yet AUTOVON network management personnel are the only near-term source of manpower and experience to solve DSN problems.

It was recognized by DCEC in FY86 that the Call-by-Call Simulator (CCSIM), a powerful DSN simulation tool built by Lincoln Laboratory during DCEC-sponsored work ending in FY85, could provide a means for resolving these problems. The original purpose of CCSIM was to study and evaluate candidate call routing and preemption techniques for the DSN, and it was a large (27,000-line) software system which had a variety of features well-tailored to that purpose. If suitably modified and extended, CCSIM could become an experimental testbed for learning to do DSN network management. Since it is very important to make this new knowledge available to field personnel quickly and completely, the notion of capturing it in a transportable Network Management Expert System became attractive. Accordingly, the FY87 program described in this report was established, with three major goals:

(1) Implement the new features and capabilities needed to suit CCSIM to these new purposes;

(2) Conduct a variety of experimental exercises with CCSIM to derive DSN network management methodologies; and

(3) Develop a prototype Network Management Expert System to serve as a repository of this near-term knowledge as well as future skills and techniques to be derived from both simulated and actual DSN experience.

1.2 Executive Summary

All the required new CCSIM features and capabilities have been implemented during FY87. The list includes:

Network Size and Description Improvements

Engineered Routing

Trunk-Oriented Rather than Switch-Oriented Routing

In-Band Signalling

Line Busy Calls

Network Management Controls

Switch Reports

Improved Statistics

Interaction with Expert System

Many of these features, such as engineered routing and in-band signalling, were necessary to match CCSIM to the current transitional DSN. Other additions included simulated data sources (namely switch statistics reports) representing the information that will be available to DSN network managers, and the network management control options expected in the near-term DSN, as well as facilities for observing the results of control actions. The control options addressed include:

Code Blocking (CB)

Call Gapping (GAP)

Cancel-to-Link (CANT)

Skip-Link (SK)

Directionalize (DRZ)

Alternate Route Cancellation (ARC)

The last of these has not yet been fully implemented. Finally, since this new application of CCSIM requires more and different statistics outputs than the original, a set of new output formats and options was developed.

To serve as a basis for simulation efforts, a wide-ranging set of generic DSN crisis and damage scenarios was created, as described in Section 3. The attributes of these scenarios were based upon study of network topologies and the associated vulnerabilities and weaknesses, focusing initially upon a DCEC-supplied 23-node model of the Pacific DSN backbone. The scenarios included single- and multiple-switch outages and trunk outage categories, and the total number of possible combinations is enormous. Simulation activity focused upon a smaller number of representative examples, and detailed results are presented in the body of the report. Examples will later be selected from scenarios that have not yet been used, in order to test the validity of the DSN NM knowledge base that has been generated.

As a criterion for evaluating and comparing various network management strategies to recover from these damage scenarios, we have recognized that it is not appropriate to focus exclusively on Grade of Service (GOS), the traditional measure of switched voice network performance. As explained in Section 4, a primary reason for this inadequacy is that GOS gives no

indication as to whether excessive preemption is occurring. A second problem is that GOS can be arbitrarily reduced (i.e., improved) by simply cancelling traffic to an arbitrary degree. Accordingly, we have proposed two new measures of network performance called "Call Failure Rate" (CFR) and "Mean Tries for Success" (MTFS). CFR is defined as the fraction of call intentions that ultimately fail to succeed (in spite of retries to overcome blocking or preemption) in completing the desired number of call seconds to the intended destination. MTFS is defined as the average number of attempts (including retries) needed to ultimately succeed in completing the intended duration of successful calls. We have taken as our primary NM goal the minimization of CFR for calls that have some hope of success (i.e., for calls not aimed toward a destination that has become unreachable due to network damage). As our secondary goal we take the minimization of MTFS and GOS, subject to the constraint that CFR is not significantly increased. An additional constraint which must be recognized is that the new measures can be used effectively in CCSIM-based research on NM techniques, because CCSIM allows access to all the necessary statistical information; CFR and MTFS cannot be used in real network operations, however, since it is impossible to compute them from the limited information normally supplied in switch statistics reports, the only source of network performance data that will be available to DSN network managers.

Among the experimentally derived NM observations we have noted, as discussed in detail in Sections 4.6 and 4.7, are:

(1) The choice of appropriate NM control action to correct a given problem is far more dependent upon current traffic patterns than upon particulars of the damage situation.

(2) Application of 100% Code Blocking to destinations made unreachable by current network damage is always a good NM action.

(3) There are traffic congestion situations that would be better addressed by altering network routing behavior than by applying NM controls. We think of such measures as "expansive" techniques because they offer new options rather than limiting traffic.

(4) Under damage scenarios in which offered traffic still exceeds available network capacity even after all expansive NM controls have been applied as appropriate, modest additional network performance improvements can be realized by judicious application of restrictive controls such as Cancel-To, partial Code Blocking, or Call Gapping.

(5) Restrictive NM controls are very sensitive to traffic levels and patterns, and their application requires careful monitoring to avoid worsening damage effects instead of alleviating them.

Based upon simulation experience to date, our recommendations for NM control actions to be taken in the event of DSN-PAC network damage are:

(1) For switch damage other than gateways, apply Code Block (CB) to destinations reachable only through the damaged switches. [For gateway damage, start with (2) below. For trunk outages start with (3) below.] While waiting for these controls to take effect, do nothing.

(2) When the CB controls (if any) are in place, put 100% SK controls on all the trunk groups to the damaged switches.

(3) Examine switch reports for evidence of excessive blocking. Overflows from trunk groups are to be expected in damage scenarios

and heavy traffic situations. They are not worrisome unless there is evidence of excessive blocking. If none is found, there is no need for further control action, but the situation should be carefully watched so that if traffic increases, action can be taken. If excessive blocking is present in parts of the net, (4) and/or (5) may be applicable.

(4) If the switch reports show evidence of heavy overload, it is unlikely that significant help can be gained by attempting to use trunk resources that are not fully utilized. In such a case, proceed to (5). If the reports show only moderate overloading with the traffic of the moment, and some resources are not being fully utilized while others are saturated, apply Skip (SK) controls or routing table changes, if possible, to make use of the spare trunk resources. These actions should only be undertaken by experienced personnel who understand the operation of the routing algorithm in the network. Serious damage to network performance could result from inappropriate routing table changes.

(5) If heavy overload conditions exist, or moderate overloads remain after any expansive controls under (4) have been implemented, the use of restrictive controls such as CANT should be explored. [We expect that Alternate Route Cancellation (ARC) will also be applicable, but we have not yet explored its use in any experiments.] CANT should be applied to overflowing trunk groups that are the last groups in routing chains. In CCSIM switch reports, we show such end-of-chain overflows explicitly. If they are not available in the real network's switch reports, knowledge

of routing tables together with assumptions about the traffic can be used to infer which overflows are likely to correspond to end-of-chain. The number of out-going incomplete call attempts should correlate with end-of-chain overflows. We recommend use of the gapping flavor of CANT with the parameter set so that the overflows remaining after the control is in place will be larger than the number of calls cancelled by the control. Simulation results show that cancelling enough traffic to bring the overflows to zero will result in excessive call failures. The idea is to cancel as little traffic as necessary. Unfortunately, we do not yet have a formula relating observed GOS and/or trunk group overflows to CFR so that a precise recommendation for parameter setting might be made. Our experiments show that the best results are obtained with a ratio of 2 to 4 between the overflows and the number of calls cancelled. The recommended procedure is to pick a setting, observe the result, and adjust as required to keep the cancelled calls within the suggested range. That range depends upon the retrying behavior of the users. On the average, a cancelled call removes n call attempts from the network, where n depends upon the average retry persistence of the users.

Our experiments show that if the expertise to follow procedures (4) and (5) is not available, or the switch reports are not coming through to the controller, the consequences of omitting those actions are not severe. The benefits to be expected from these procedures are typically the saving of a few percent of the call intentions. The penalties for misuse are likely to be an equal or greater increase in the failure rate.

The following two general recommendations reflect additional observations we have made about application of NM controls:

(6) If NM controls that cancel calls (partial CB, GAP, and CANT) are to accomplish their purpose, it is important that callers receive a message that effectively prevents retry of the call in the event that a call is affected by the control. A message or signal interpreted by the caller as equivalent to a blocked call indication will result in the control action having no useful effect.

(7) Retrying persistently is important for achieving a useful CFR in a damaged network. Therefore, we would recommend against administrative actions to limit retries for damage scenarios no more severe than simultaneous two-switch outages. On the other hand, administrative action to discourage unnecessary use will always help.

Two Appendices are provided in the report: the first is a description of procedures for operating CCSIM, and the second is a description of the prototype Network Management Expert System. The latter includes an estimate of the magnitude and complexity of the work of fully implementing an experimental Expert System for this purpose, as well as a similar assessment of the effort required to extend this technology to a field-deployable Expert System that could be used operationally at DCS network management centers. We have estimated the manpower requirements for these purposes as four man-years and an additional five man-years, respectively.

2. THE DSN CALL-BY-CALL SIMULATOR (CCSIM)

2.1 FY85 Delivery System

At the conclusion of the DCA-sponsored EISN program in September 1985, a Call-by-Call Simulator (CCSIM) was completed and delivered to DCEC. This FY85 CCSIM had been written over a period of three to four years to support study of new candidate routing and preemption algorithms for the 1990s-era DSN, and its capabilities were matched to that purpose. CCSIM was written in RATFOR (Rational Fortran) to enhance portability, and consisted of about 27,000 lines of code. Besides serving the needs for which it was implemented in EISN, CCSIM has been used for a variety of purposes at DCEC since its FY85 delivery.

The general structure and capabilities of CCSIM were recognized in late FY86 as appropriate for the development of Network Management methodologies for the DSN. It was clear that it offered a long head start, and that major savings in manpower could be achieved by taking advantage of it. In a number of important respects the FY85 version was not completely matched to the task, however, and a major effort (now complete) was undertaken in FY87 to implement the needed extensions and modifications. For example, CCSIM assumed that common channel signalling (CCS) and DAMA satellites were in use, and it did not provide for in-band signalling or point-to-point satellite trunking. Hence it could not be used to simulate the current transitional DSN environment. It was oriented toward node-centered activities such as call processing which are relevant to routing algorithm studies, and did not represent trunk-oriented activities such as timing details and the tying up of resources during call routing attempts, which are relevant to network management. It did not include

network management controls, nor the gathering and reporting of five-minute switch statistics. It did not incorporate the Engineered Routing procedures that are being used in the transitional DSN. Conversely, CCSIM contained a number of features which will not in fact be implemented in the DSN, including the AUTOVON Polygrid routing algorithm and experimental algorithms such as Flooding. Since these features unnecessarily consume time and memory, they were removed.

This section of the report gives details of the work done to create the FY87 CCSIM, accurately modeling the transitional DSN for network management purposes.

2.2 FY87 Capabilities and Status

An important aspect of the current effort was removal of all the unnecessary features for the current task. The code taken out has been preserved, of course, and relevant portions of it can be restored to CCSIM in the future as the DSN evolves; however, care will have to be taken to integrate any such restorals with all the changes made in CCSIM since the removal. Meanwhile, we avoid the continual manpower drain of maintaining and updating code that is not needed. Numerically, the process of removing DAMA satellites and earth stations, Polygrid routing and flooding shortened CCSIM by approximately 9,000 lines of code.

2.2.1 New Features and Capabilities

The following is a list of the objective capabilities added to CCSIM in FY87. These objectives have been substantially achieved, and are discussed (not necessarily in the same order) in the following paragraphs.

Network Size and Description Improvements

Engineered Routing

Trunk-Oriented Rather than Switch-Oriented Routing

In-Band Signalling

Line Busy Calls

Network Management Controls

Switch Reports

Improved Statistics

Interaction with Expert System

The internal tables in CCSIM have been reworked to allow up to 512 switches in a simulation. The size and topology of the network for a given experiment are provided at run time, along with other information such as a matrix of the average busy-hour traffic between each pair of switches; the size, type and destination of the trunk groups connected to each node; a file giving the name and related information about each node; and a file containing all the engineered routing tables for the network.

For large simulations the upper bound on the number of switches in a simulation is more likely to be imposed by the available memory size, rather than any limits in CCSIM. Running time will also increase with network size. For the present we are modeling the 23-switch PAC backbone, which provides ample complexity and realism for our current network management knowledge engineering work. Typically CCSIM runs this model faster than real time, by a factor of 4 or so, on SUN work stations.

The range of network management controls expected to be available in the near-term DSN has now been added to CCSIM, as described in the following section. CCSIM now reflects the effects of calls to busy destinations and models the tying up of resources by failed calls. Switch statistics reports are now generated every 5 minutes for every switch in

the net, reflecting the best available information about the reports that will actually be produced by DSN switches in the field. Considerable tailoring was done to the overall network statistics reports generated by CCSIM, to better meet our needs in analyzing network management issues.

2.2.2 Structural Changes

During the past year, CCSIM has been modified to accurately represent the current transitional DSN having in-band signalling and engineered routing, with both terrestrial and point-to-point satellite connectivity. In addition, the near-term DSN network management controls have been implemented; network damage simulation has been improved; and two methods of interfacing with the simulator have been implemented. To support the above visible changes the internal structure of CCSIM has been modified to be link-oriented rather than switch-oriented. The change to link orientation was necessary in order to be able to represent both terrestrial and point-to-point satellite connectivity, to implement link-related controls, to improve network damage simulation, and to get the correct information for switch reports.

The major changes necessary for in-band signalling were lengthening of signaling times and curtailment of some of the routing options. Call setup time is longer with in-band signalling, including longer trunk occupancy even for calls that are never completed. The crankback routing option formerly available depended upon call path information in the CCS call setup messages; this option is no longer available, since in-band signalling does not allow for inter-switch messages.

Implementation of engineered routing required addition of an engineered route digit to specify the number of remaining routing

alternatives for the call at each switch. The set of engineered routing tables for the PAC backbone switches, including the engineered route digits, was provided to us by DCEC. The CCSIM code used formerly to create and use routing tables without engineered route digits still exists in the simulator, but is not being used at this point. Care would have to be taken to verify that this code is still compatible with all the other changes that have been made in CCSIM, before it could be confidently reactivated.

CCSIM provides for both damage and restoration of switches and trunks, as specified by the user. It is reasonable to anticipate two generic kinds of switch failures, namely functional disruption and physical damage. Functional disruption means sudden refusal to provide services normally expected, such as acceptance, processing and connection of new calls or tandem routing requests, and could correspond (for example) to loss of the CPU program. Physical damage would result from fire, flood, explosion or other disaster, and would destroy calls in progress as well as causing total functional disruption. In the current CCSIM implementation switch damage is assumed to be functional disruption, in which calls in progress are allowed to complete normally but all new requests are blocked. Physical destruction of the switch could be simulated by applying both a switch damage event and simultaneous destruction (as defined below) of all trunk capacity connected to the switch.

When switch damage is commanded during a CCSIM run, an internal flag is set indicating that the condition exists. As noted above, calls in progress are not terminated. The neighbor nodes of the damaged switch are not informed that the damage has occurred, but are left to discover the

situation gradually (as they would in the real world) by failing to receive signalling response each time they attempt to route a call to the damaged switch. For each such attempt, after a preprogrammed time-out the calling switch concludes that the trunk being accessed is bad, and proceeds to decrement its figure for available link capacity to the damaged switch. Eventually, unless the damaged switch is restored as explained below, all the neighbor nodes will see zero capacity on all links to the damaged switch.

When restoral of a damaged switch is commanded during a CCSIM run, we assume (again reflecting the most reasonable real-world approach) that neighbor nodes are so informed, and they reset link capacities to the values they had before the switch was damaged. When trunk damage is commanded, the capacity of the associated link is reduced correspondingly, and enough calls are removed from the link so that the number of busy trunks does not exceed the remaining capacity of the link. By removing calls in this way, we are making the assumption that the first links we consider damaged are the free links. This produces the worst-case scenario; in other words, it will cause the highest incidence of near-term blocking. The long-term impact of exactly which trunks are damaged and how many calls are removed has little effect on the simulation results. Finally, when trunk restoral is commanded the capacity of the link is increased correspondingly, and the switches at each end of the link are so informed.

During FY87 five network management controls have been implemented: code-block (CB), call-gap (GAP), cancel-to-link (CANT), skip-link (SK), and directionalize (DRZ). In fact a sixth control (alternate route cancellation, or ARC) has been implemented, but debugging is not yet complete.

The first two controls are used to limit or prevent attempted traffic to a node known to be congested or damaged. A CB message received by any node in the network will cause it to cancel a fraction in eighths (from 0 to 8) of all call attempts from itself to the damaged node. A GAP message to a node will cause it to impose an upper limit on the rate (in calls per minute) of call attempts from itself to the damaged node. Both controls are applicable to Routine calls only, unless 8/8 of the calls are canceled or the highest level of gapping (which also blocks all calls) is applied. When these highest levels of cancellation and gapping are used, all call precedence levels are affected.

The other three controls implemented are link-oriented; they are applied at only the neighbor nodes of a damaged node, or at both ends of a damaged link. A CANT message can be sent in either of two available versions: one of them causes Node 1 to cancel a specified fraction of its attempts to route calls on its link to Node 2, while the other version imposes an upper limit on the rate of such attempts. An SK message seeks a similar goal, but without automatically canceling the call: for a specified fraction (in eighths) of call attempts that would have tried to access the specified link, Node 1 is directed to skip the corresponding line in its routing table. The next routing table entry can be used if available to complete the call. The DRZ control can be used to give preferential treatment to Node 1 on its link to Node 2, by reserving a certain number of trunks for seizure in the direction from Node 1 to Node 2. DRZ can be used in a focused-overload situation to improve the chances of callers at the overloaded node to get out to the rest of the network.

As noted above, two new methods have been implemented for interfacing with CCSIM. One is an initial version of an interactive graphics program, and the other is a new command file input to CCSIM. The interactive graphics program is a menu-driven facility which allows the user to run or stop CCSIM, display offered and blocked call matrices, apply and remove network damage, and issue controls. It normally displays a topological map of the network, and it replaces the map with the call matrices when requested by the user. The graphics facility allows the user to react to network damage scenarios at run time, rather than being required to pre-plan all his damage and control commands.

In the command file method of running CCSIM, the user creates a file in advance which represents the entire sequence of damage, restoration and control events. This file is read in while CCSIM is running, with TIME commands specifying the simulated time at which each group of one or more damage or control events takes place. Any of the controls and damage options discussed above can be applied through the command file. In addition, commands are available for controlling switch report output. The main advantage of the command file method of running the simulator is that many variations on a scenario of interest can be run by making simple changes in the file. Also, it permits the user to do runs in batch mode while he is busy with other things.

2.2.3 Statistics and Switch Reports

CCSIM produces tables and statistics to aid in the analysis of the runs made. These statistics attempt to provide the facts needed to analyze each run. In addition to the statistics provided by the FY85 version of

CCSIM, several plots and additional statistics have been added. New statistics and/or plots will be added as the need for them becomes apparent.

Information on the number of calls which were tried N times and their GOS has been expanded. For each value of N the number of calls blocked, preempted and dropped is included. This set of statistics is given for the total number of calls in the interval and then broken down by precedence level. A new category giving the total number of calls encountered in the various categories since time zero has been added. An example of these tables is included in Appendix A.

The most useful addition to the CCSIM statistics outputs has been a set of four matrices for each precedence level. (If there are no nonzero values, the matrix is replaced by a one line statement.) The matrices presented evolved as the need for the information became apparent. The first matrix, labelled "BLOCKING MATRIX FOR PRECEDENCE LEVEL X", shows the GOS between all possible source-destination pairs for that precedence level. To conserve space the GOS is multiplied by a thousand and printed as an integer between 0 and 999. (A GOS of 1.000 is printed as 999 in order to hold to three printed digits.) In order to evaluate the meaning of any GOS it is necessary to know the number of calls involved. Therefore two more matrices were added. The second matrix, labelled "BLOCKED CALL MATRIX FOR PRECEDENCE LEVEL X", records the number of blocked calls between each switch pair for precedence level X. The third matrix, labelled "OFFERED CALL MATRIX FOR PRECEDENCE LEVEL X", shows the total number of calls placed between switch pairs for precedence level X. When the code

cancellation and code gapping controls were added to CCSIM, a fourth matrix was added to show the number of affected calls between any two switches for the indicated precedence level. This matrix is labelled "CODE CANCELLED OR CODE GAPPED MATRIX FOR PRECEDENCE LEVEL X". This information is very helpful in analyzing problems in the net. Examples of these statistics matrices are included in Appendix A.

Since CCSIM starts with no calls in progress it must run for a considerable length of time before a steady state performance level is reached. Meaningful statistics cannot be gathered until then. The user can specify this point in time by setting an initialization parameter called TVALID in the INVAL file. At TVALID there are many calls in process in the net using resources. There are also preempted, blocked and line busy calls waiting to retry. This complicates the analysis of statistics from a slice of time following TVALID. Currently, the number of calls on the retry list at TVALID is included in the statistics. We expect, however, that more information will be needed to adequately reflect the state of the net at TVALID; we plan to provide it in future versions of CCSIM. Similarly, as it becomes apparent that further statistics would be useful, they will be added to CCSIM whenever practical.

DSN switches are expected to compile and send out reports every five minutes. CCSIM now sends two such reports. The first report, referred to as the EXPERT report and contained in a file identified by the tag .exp, is intended to contain the information sent out by an actual switch in the network. The second report, referred to as the SWITCH report and contained in a file identified by the tag .sw, is designed to provide data for

analysis by those evaluating damage scenarios and the effects of controls applied.

The EXPERT file contains a subset of the information listed in Section 16.3 of TR-TSY-000064 LSSGR December 1984 entitled NETWORK, expanded to deal with precedences. In addition, when a control is implemented in CCSIM, a report of the number of calls affected by that control is added to the EXPERT file. The EXPERT file will be sent to the Expert System Program. These reports are the only source of information available to the Expert System program as it monitors the state of the network and applies controls as indicated. The format of the EXPERT file has been devised so that it can be easily parsed by the Expert System. It contains values in known order with some labels. It is important that this file be as compact as possible, to minimize the load on statistics communication links and on Expert System pre-processing facilities. Appendix A includes an example of an EXPERT file.

The SWITCH report contains all the information which is in the EXPERT file in a more "user friendly" form. Values are labelled so that they are more easily understood. Some values that can be calculated from values in the EXPERT file are added to the SWITCH report to help the human analyzer. Since this file is only examined locally, the extra length is not important. Appendix A includes an example of a SWITCH file.

In both reports all appropriate values are reported by precedence level. The figure for Usage (traffic over a trunk group) is not currently reported by precedence. For the final system it is assumed that reports from various nodes will not be synchronized. Therefore on initialization

of CCSIM, a time for the initial report of each switch (from zero to five minutes) is chosen at random. After the initial report each switch will report at five minute intervals. To aid in debugging during early experiments, CCSIM currently overrides this initialization and synchronizes the reports. Synchronized reports also aid in the analyses of the results obtained when controls are applied. A sample of a SWITCH report from two switches and the corresponding EXPERT report is contained in Appendix A.

The totals following each SWITCH report can signal various problems in the net. If none of the trunk groups or switches that a given node is connected to is damaged, the total number of calls routed out of that switch should equal the total number of calls sent out over its links for precedence level one calls. For example, if the switch at the destination end of a trunk group fails, the source switch does not get an in-band signal response when a call is routed over the trunk group. This causes the source node to declare that trunk in the trunk group inoperative. While the source node is discovering one by one that each trunk in that trunk group is inoperative, the total of calls routed out of the switch will exceed the total of calls sent out over the links. Finally, when all the trunks in the group have been declared inoperative, the totals will agree.

3. DSN CRISIS AND DAMAGE SCENARIOS

3.1 Objectives

We have three objectives in developing crisis and damage scenarios. The first is to provide a wide-ranging set for use in learning about the effectiveness of network controls in the current and future DSN and for training the expert system we are developing. The second is to provide a

different set for testing the effectiveness of the expert system. The third is to provide a basis for recommendations for NM control actions that would be applicable in the near-term developing DSN.

Since we are not operating in an appropriate classified environment, we cannot make use of information about the location of critical network users and consequently cannot develop true crisis scenarios or explore vulnerability issues. We therefore have concentrated on damage situations and their impact on normal busy-hour and other arbitrary traffic patterns. However, we believe that the tools we are building will prove useful for vulnerability studies and crisis scenario investigations when they can be run in an appropriate classified environment.

3.2 Rationale

A complete damage scenario specifies some sequence of damage events with associated traffic pattern changes. The damage events should include both the loss and recovery of network assets so that the effect of network controls can be observed in both cases. Our work to date has not progressed to the point of generating complete scenarios but has been limited to working out sets of damaged network states which can be used as components of complete scenarios.

In working out the damage states, we have found it useful to think of DSN-PAC (Fig. 3.1) as made up of seven geographical regions (Table 3.1).

Table 3.1

Code	Name	Region/Function
CKS	Clark AB	Phillipines
CON	CONUS	-- Fake node --
CRC	Camp Red Cloud	Korea

DSN-PAC BACKBONE NETWORK

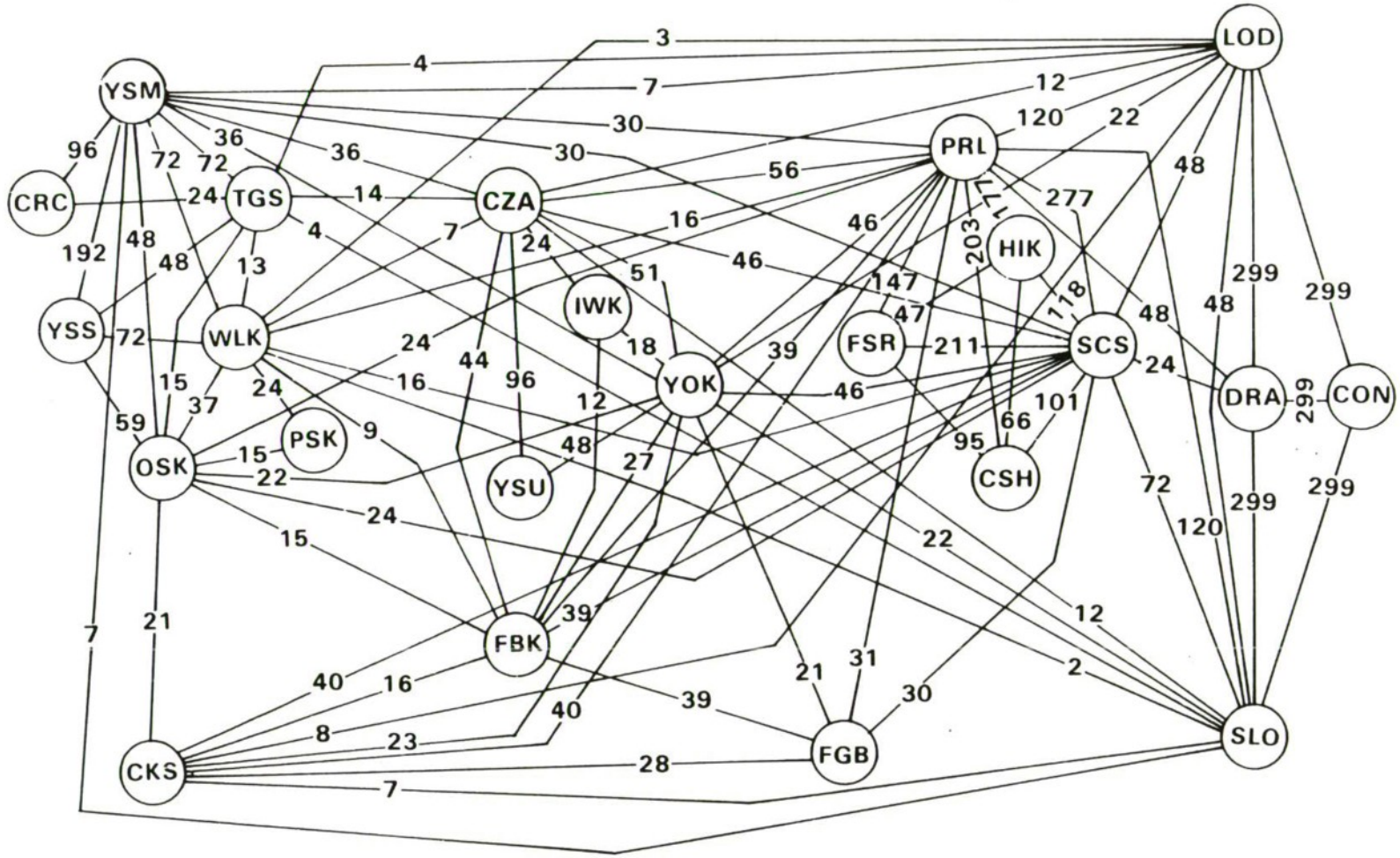


Figure 3.1.

CSH	Camp Smith	Hawaii
CZA	Camp Zama	Japan
DRA	Dranesville	CONUS - special
FBK	Fort Buckner	Okinawa
FGB	Finegayan Bay	Guam
FSR	Fort Shafter	Hawaii
HIK	Hickam AFB	Hawaii
IWK	Iwakuni	Japan
LOD	Lodi	-- CONUS gateway --
OSK	Osan AB	Korea
PRL	Pearl Harbor	Hawaii
PSK	Pusan	Korea
SCS	Schofield Barracks	Hawaii
SLO	San Luis Obispo	-- CONUS gateway --
TGS	Tango	Korea
WLK	Camp Walker	Korea
YOK	Yokoto AB	Japan
YSM	Yongsan Main	Korea
YSS	Yongsan South	Korea
YSU	Yokosuka	Japan

Three of these - Guam, Okinawa, and the Phillipines - have only one multi-function or stand-alone switch. A fourth - the continental United States - is a special case since it is another network with its own routing algorithms and responses to damage and overload. The CONUS-DSN is connected through gateways to DSN-PAC, and scenarios involving the gateway

switches must consider the behavior of the CONUS network on traffic destined for DSN-PAC. A further special case associated with CONUS is the DRA node, which looks on a network map like a gateway between DSN-PAC and CONUS but is not treated as such by the routing tables. For damage scenario studies DRA looks like yet another single-switch region.

The remaining three regions - Hawaii, Japan, and Korea - are multi-switch regions with one or more switches that have no inter-regional trunks. For these regions we define three switch types - primary, secondary, and tertiary - according to the number of inter-regional trunks nominally available to the switches. Some tertiary switches (e.g, CRC and PSK in Korea) have the further property of offering no tandem functions for other calls in the network. In addition, the Korean region as a whole has the property that none of the switches provides tandem service for calls whose source and destination both lie outside the region.

As far as damage events are concerned, we have considered switch outages to be total but trunk outages to be both partial and total. There are two reasons for restricting our scenarios to total switch losses. The first is that we do not have a model for switches in CCSIM that can provide any useful simulation of partial switch failure. The second is that our preliminary investigations of switch capabilities indicate that the new switches being procured for DSN-PAC will have sufficiently high capacity that machine congestion should not occur with the number of stations and trunks that are currently planned. It is of course possible for a switch to experience partial failure due to hardware malfunction or inadequate provisioning of resources, but it is not clear how to relate such failures

to switch performance vis-a-vis the handling of inter-switch calls which is all that CCSIM models.

The set of damage scenarios should contain both single and multiple switch and trunk outage categories. In the single switch failure category we see the following sets:

1. Primary switch in the region - PRL, YOK, YSM, FBK, ...
2. Secondary switch in the region - SCS, CZA, OSK, ...
3. Tertiary switch - FSR, IWK, CRC, ...
4. CONUS gateway - LOD, SLO

In the multiple-switch failure set there are a vast number of combinations to explore. Looking just at two-switch failures, we see the following groupings:

1. Different regions in DSN-PAC - PRL + CKS, YOK + SCS, YSM + FGB, ...
2. Gateway plus DSN-PAC switch - LOD + SCS, SLO + YSM, ...
3. Inter-regional switches - PRL + SCS, YOK + CZA, YSM + OSK, ...

It should be noted that scenarios involving gateways require adjustments to the traffic matrix to reflect the compensation that the CONUS network would make for traffic coming from CONUS to DSN-PAC.

There is no problem in running damage scenarios with CCSIM that involve any number of nodes up to the entire network. In one sense, the network management task becomes simpler as the number of damaged switches increases since there are fewer resources to monitor and control. In another sense it becomes more difficult because the routing algorithm is less likely to make the best use of the remaining resources. We also expect that the goals of NM actions would be different in cases of severe overall network damage. For example, it seems likely that NM personnel

would focus on trying to maintain service for certain critical users at the expense of others when resources are severely reduced, whereas in a less pressing situation, they would be more concerned with fairness. Our understanding of the issues involved is not yet sufficient for us to identify any particularly interesting multiple-switch damage scenarios.

In contemplating trunk outage scenarios we are working with the following dimensions in selecting a scenario:

1. Inter- vs. intra-regional trunking.
2. Satellite vs. terrestrial trunks.
3. Large vs. small trunk groups.
4. Partial vs. full outages.
5. Single vs. multiple outages.

Multiple outages can again be partitioned a number of ways. Among the possibilities are the following:

1. Unrelated outages at random points in the network.
2. Multiple outages at a single switch or within a region.
3. Particular outage patterns caused by damage to a satellite, earth terminal, or commercial facility that might be common to trunk groups that appear unrelated on a casual examination of network topology.

Trunk outage scenarios generally have the property that resources are lost without any compensating loss of traffic. They therefore may show a greater effect on network performance than would be observed in a damaged switch situation where a fraction of the network load goes away with the resources. The extreme case of the latter is found with switches such as CRC and PSK in Korea that have no tandem functions. If traffic to such a switch is cancelled at its sources, the loss of the switch will have no

effect on network performance as seen by other traffic. Since there are no switches in the backbone DSN-PAC with only a single trunk group to the rest of the backbone, there is no case where a single trunk outage will not affect other traffic.

In our view, some damage events could be expected to occur under normal traffic conditions while others would be accompanied by unusual traffic stresses on the network. The first case could occur in the real world due to chance failure of some network resource. The latter case would occur if the damage were due to hostile action or natural disaster (e.g., storm or earthquake). Unfortunately, real world models for such abnormal traffic are not available. All we can do is to approximate such cases by introducing arbitrary focused overloads both within the region where the damage exists and in other areas (e.g., traffic to and from CONUS through the gateways). We have not as yet conducted any experiments in which we attempted to model abnormal or focused-overload traffic in a "realistic" way. We have done some experiments in which traffic to and from a region (Hawaii, to be specific) was increased by a factor of two relative to the traffic in the remainder of the net just to see what would happen, but such a simple scenario is not very useful since it is too obvious what controls should be applied to compensate for the overload. A more realistic scenario would apply a less uniform distribution to the abnormal traffic.

We have been ignoring timing in our initial experiments and have considered damage events to be steady-state situations. Until the interactive graphics front end for CCSIM is developed to such a point that displays can show what is going on in the simulated network in a dynamic

fashion, it is simply too cumbersome to attempt analysis of the transient effects of damage and control applications. The human observer is overwhelmed by the mass of detailed data generated by CCSIM. As the interface develops, we expect to investigate the dynamic behavior of the net and be able to address the issue of defining meaningful event timings for scenarios.

3.3 Scenario Selection

For use in learning about the effectiveness of NM controls, for training the expert system, and for providing data for near-term recommendations, we have selected a number of scenarios using the categories described in the preceding section. We have not yet selected a set for testing the expert system, but since the number of scenarios in most of the categories is quite large, there is no need to take pains to leave some unexplored for later testing of the expert system. Only in the single-switch failure category is it neccesary to be careful so as to leave some scenarios untried until testing time.

To date we have examined the following scenarios in the indicated categories:

Single Switch Damage

1. Primary switch in region - PRL, FBK
2. Secondary switch in region - TGS
3. Tertiary switch - YSU
4. CONUS gateway - LOD

Multiple Switch Damage

1. Different regions in DSN-PAC - YOK + SCS
2. Gateway plus DSN-PAC switch - LOD + SCS

3. Inter-regional switches - YSM + OSK

Trunk outages

1. Intra-regional, large trunk group - YSM <-> OSK (48 trunks)

2. Inter-regional, small group - CKS <-> SLO (7)

3. Partial outage of large group - PRL <-> LOD (120 reduced to 100)

As work progresses we will expand this list as needed to support our knowledge engineering work for the expert system.

4. NETWORK MANAGEMENT EXPERIMENTS

4.1 Objectives

We have three objectives for our network management experimental activities. One is to provide the knowledge needed for the expert system we are developing. In conventional expert system development the knowledge is obtained through a process called knowledge engineering that involves intensive interaction with human experts in the field of interest. In our case, because the DSN in not yet operational in its targeted form, there are no experts with experience in managing it. There are, of course, people with experience in managing commercial telephone systems and the earlier generations of military networks, and we are making use of their knowledge to the extent that it is available to us. However, we are ultimately dependent upon knowledge derived from CCSIM runs in formulating NM rules to be implemented in our expert system program.

The second objective is to provide recommendations for NM control actions that might be applied in the evolving DSN during the period when experience with the real network has not yet provided example scenarios from which controllers could learn about the effectiveness of the available NM control options.

Our third (and longer-range) objective is to evaluate and compare the effectiveness of the available control options, alone and in various combinations, under various damage scenarios.

The work we discuss in this section is based on the analysis of approximately three hundred simulator runs. The runs have made use of two different versions of routing tables and three different sets of traffic and link capacity files provided by DCEC in February, May, and October 1987. The bulk of the data from these runs is too great for in toto presentation of even the summary data. Further, since the runs were carried out over a period of time during which CCSIM development was continuing and the network definition was changing, it is not always valid to make direct comparisons between numbers obtained in different time periods. Consequently, we will present here only a small number of representative results to illustrate particular points relative to our objectives. In cases where we are comparing numerical values, we have been careful to make sure that the runs used the same version of CCSIM and, except where noted, the same network definitions.

Precedence traffic is unblocked in all the examples tested, except in a two-switch damage case described in Section 4.5.5, where precedence call blocking was caused by an apparent design error in the routing tables.

4.2 Network Management Goals

With respect to damage and overload scenarios we assume that the goal of network management is to apply NM controls to improve network performance. There are other network management actions such as rearranging or patching transmission resources that may be indicated in dealing with damage scenarios, but these fall outside the scope of our work and are not addressed in this report.

The concept of improving network performance seems clear enough at first thought but proves to be rather fuzzy on further consideration. In fact, we have concluded that network performance under damaged conditions is not a well-defined concept and have felt compelled to make a definition that we can use in comparing one control procedure with another. The problem comes down to deciding what we will measure to decide whether an action benefits or compromises performance. To start with, we have the traditional measure called "Grade-of-Service" (GOS). It is used along with traffic estimates to design the network routing tables and trunk sizing. It measures the probability that a call attempt will be blocked, i.e., no path between source and destination can be found for the call. (In such a case the caller receives either a fast-busy signal or a recorded announcement to the effect that all trunks are busy at the moment.)

As a direct measure of network performance, GOS has two serious problems for use in the DSN. One is that it does not give any indication of whether or not preemptions are occurring at excessive rates. For example, there can be situations in which GOS indicates a low probability of blocking and callers are getting through readily but are very soon being preempted with a resulting need to try many times to complete an intended conversation. The second problem with using GOS as a primary measure is that it can be arbitrarily reduced (improved) by cancelling traffic into the net. In the extreme case we can get a perfect GOS (0.0) by cancelling nearly all the offered traffic. We therefore cannot take the minimization of GOS to be the primary goal of NM in a damage or overload situation. However, other things being equal, reducing the value of GOS is a good thing to do.

Our problem now is to decide what other things we want to keep equal. Our proposed candidate for this role is a new measure we call "Call Failure Rate" (CFR). We define CFR to be the fraction of call intentions that fail to achieve and maintain connection for the duration of the intended call. It takes preemptions and retries into account, and includes calls cancelled by network controls as failures. We believe CFR provides a realistic measure of the utility of the network to its users. (We could equally well have defined a "Call Success Rate" that would have a value of one minus CFR, and some people to whom we have presented these ideas have indicated a preference for that success-oriented measure. However, since GOS is an established measure in which better performance gives a lower value, we have decided to stick with CFR which also has the lower-is-better property.)

Unfortunately, CFR has two problems that probably prevent its use in the real world but not in our simulator experiments. The first is that its actual value depends upon the retry behavior of the callers. Greater persistence in retrying tends to improve (reduce) CFR, lesser to worsen (increase) it. The second problem is that it is not directly derivable from the switch report information provided to network controllers. We hypothesize that CFR can be estimated from switch reports by combining trunk overflow and preemption data with an assumed retry model, but our work has not progressed to a point where that hypothesis can be either confirmed or denied. In any event, it appears to be a useful measure in our simulation experiments where it can readily be computed from CCSIM outputs.

A third measure we have been using is called "Mean Tries for Success" (MTFS). It is computed by calculating the average call attempts needed to

complete a successful call as defined in the preceding paragraph. The exact value of MTFS, like CFR, depends on the retry behavior and is not readily computed from switch reports. Its utility lies in our simulation environment, and we do not foresee a need for its estimation from switch reports.

Given the above-described measures of network performance, we take as our primary NM goal the minimization of CFR for calls that have some possibility of success. As a secondary goal we take the minimization of MTFS and GOS without significantly increasing CFR or adversely affecting critical users. (It appears that once calls that cannot succeed have been cancelled in a particular scenario configuration, MTFS and GOS are strongly correlated, and independent control actions are not needed to effect the minimization of the two measures.) As a tertiary goal for NM, we propose trying to achieve fairness by equalizing point-to-point CFR and GOS without spoiling overall CFR. (Experience to date suggests that the chances of success with this last goal are not great and are highly dependent upon the damage scenario.)

4.3 Retry Models

Since we intend to make use of measures that depend on retry behavior, it is appropriate to digress for a discussion of retry models. It is clear from our experiments that the retrying of blocked and preempted calls is an important factor in the performance of networks under stress. When underlying resources are congested, persistent retrying can cause the offered traffic to appear to increase by a significant factor with a consequent degradation in the observed GOS. At the same time, such retrying is essential if CFR is to be kept low.

The retry model currently implemented in CCSIM applies a probabilistic test after each blocking or preemption event. If a call passes the probabilistic filter, it is retried after a randomly selected interval whose mean value is a simulator run-time parameter. If the event was a preemption, the duration of the retried call will be the remaining duration of the original intended call plus an arbitrary 15 seconds that is added in the belief that the participants in the preempted call would take a little extra time to reestablish the conversation. A limit is placed on the total number of retries (preemptions plus blockings) that a call intention is allowed. CCSIM also allows the experimenter to indicate whether calls that are damaged by trunk outages should be retried or not. There are usually so few such calls in an experimental run that whether they are retried or not is immaterial.

In order to be able to compare results between different runs, it is important to use the same retry parameters on all runs unless there is particular interest in observing the effects of varying the parameters. In the results reported here we use probabilities of 1.0 for retrying both blocking events and preemptions and set the limit at nine retries. The limit is necessary to prevent memory overflow problems for calls that cannot succeed to unreachable destinations. We make no claim that this model is realistic. Obviously, no community of users would all retry with exactly a persistence of nine. Unfortunately, probabilities less than 1.0 in this model do not yield behavior that is any more defensible, and they have the serious disadvantage for our experiments of being less repeatable due to the random decision at each retry choice point.

In the future, we plan to introduce a new retry model that will pick a retry persistence value at call generation time. This behavior will allow repeatability since the call generator will produce an identical sequence of call intentions on repeated runs unless the seed for the random number generator is changed. The distribution function for persistences will use a table allowing an arbitrary shape with a guaranteed upper limit. We plan also to allow retried calls to increase in precedence up to a limit set by a class mark determined by the call generator for each intention. A possible further extension of the new model would allow the interval between retries to increase as the number of retries increased. We expect that this new model can better represent the real world (or some experimenter's idea of how the real world behaves) than the one presently in use.

4.4 Experiment Conditions

All our experiments have made use of one or another version of the DSN-PAC backbone switch network shown in Fig. 3.1. The network consists of 19 multi-function or stand-alone switches in the Pacific theatre, one special switch in CONUS, two gateway switches to the CONUS DSN, and a single fake node (CON) representing a destination for all traffic from DSN-PAC to CONUS. In the figure the numbers shown in the lines between the nodes are the total trunk sizes between the associated node pairs. These trunk sizes are those of the latest version of the network which we received in October 1987 and are currently using to calibrate our simulation results against the results obtained at DCEC with their network modelling program. In CCSIM the trunking is distributed between terrestrial and satellite trunks, but since the current routing tables make

no distinction between the types, we have combined the numbers to simplify the figure. The behavior of the network would be different if a routing control digit was used to limit the number of satellite hops, but such is not the case for the current DSN version or our simulations.

The traffic files used for the experiments are based on two files provided by DCEC. These are traffic estimates for Eastern Pacific and Western Pacific busy hours that were used by DCEC in sizing the trunk capacities and building the engineered routing tables. There are two different busy hours due to the time zone differences across the net. We manually edited the files to remove any local traffic since CCSIM does not give any useful outputs for such traffic, and its presence in the traffic file slows the simulation runs and clutters the output presentation. Traffic from CONUS into DSN-PAC in these files is distributed between the two gateway switches (LOD and SLO) according to DCEC estimates as to how the CONUS DSN would route the traffic. For gateway damage scenarios we edit the traffic files to move the traffic from the damaged gateway to the remaining gateway on the assumption that the CONUS DSN would take such action in the event of damage to the gateway.

When CCSIM runs, the traffic file specified in erlangs between source-destination pairs is converted into a sequence of calls which, if completed succesfully and held for the full duration, would place an erlang load on the network that would correspond to values in the file. For our purposes, we treat the calls as intentions and submit them to the retry mechanism described above. The retry process may substantially increase the number of call attempts seen by the network, but the erlang load is not increased except to the extent that preempted calls are extended by

15 seconds for each preemption. In all experiments, calls cancelled by control are removed from the retry process.

The call intentions follow a distribution of precedences suggested by DCEC. This distribution gives 67% routine calls with the higher precedences at 25%, 7.4%, 0.5%, and 0.1% of the call intentions. With damage in the net, the carried traffic generally has a much higher fraction of precedence calls.

Since our interest at this time is in damage states rather than transient behavior, we use relatively long run times in our experiments. In the results reported here, we run the simulator for one hour of simulated time with the traffic of interest. Then we introduce the damage event and controls (if any) and run for another hour to let the transients settle. We then turn on statistics collection and gather data for a third hour. All our comparisons are based on performance in this third hour with the same call intentions. For DSN-PAC of Fig. 3.1, busy-hour traffic levels, and single node damage, typical experiments require four to five minutes of real time to run on our SUN 3/260 workstations. Heavier traffic or more extensive damage will result in longer run times.

4.5 Example Scenario Results

In this section we present some examples of experiments involving damage scenarios. While we present only a few, each is representative of a class, and our evidence is that the conclusions extend to other members of the class.

In the examples, when we show a value for GOS without any qualifiers, we are presenting the grade of service experienced by callers who have some chance of success, i.e., they are not calling a destination that is

unreachable due to the damage.

For reference in the examples below, baseline performance for the undamaged network is:

	GOS	MTFS	CFR
West Pac busy hour	0.012	1.055	0.0000
East Pac busy hour	0.093	1.183	0.0005

4.5.1 Switch Damage - TGS (Tango, Korea)

The Tango (TGS) switch in Korea is an example of what we call a secondary switch in the Korean region. It has trunks out of the region and provides some tandem service for other switches in the region, but it is not the largest switch providing such services for the region. Other than cancelling traffic actually meant for TGS destinations, we find that controls applied in this case have litle effect on network performance. We used Western Pacific busy hour traffic in our experiments with TGS damage because it stresses the remaining resources more than does Eastern Pacific traffic.

If we damage TGS and apply no NM controls, we observe the following:

	GOS	MTFS	CFR
No controls	0.043	1.110	0.0000

Under these conditions the apparent GOS which includes the failing attempts to reach TGS is 0.185. This higher value is the GOS that would be obtained by averaging the blocking statistics from all the switch reports and is the only GOS figure accessible to a network controller. Such a difference is typical and shows that the performance of the net in this situation looks worse to the controller than it actually is.

If we follow the recommended procedure and cancel the traffic that cannot succeed via code block (CB) controls applied at the source switches, we observe:

	GOS	MTFS	CFR
CB 8/8 to TGS	0.041	1.094	0.0000

(The notation "8/8" means that the control is cancelling eight out of eight parts of the traffic, i.e., all of it, including the higher precedence calls.)

In this case no further control actions are indicated, since no calls are failing, and performance is very close to that for the baseline WP busy hour:

	GOS	MTFS	CFR
Baseline, no damage	0.012	1.055	0.0000

If traffic had been heavier in relation to the remaining resources, other control actions might have been needed. The next scenario we describe (Sec. 4.5.2) will illustrate such a case.

Another NM control question is what, if anything, should be done during the interval between damage and the application of Code Block (CB) controls? In a manual control situation such as will exist until DSN is fully operational, this period could be quite long since the CB controls are applied at end offices, and there are many such. One possible control action would be to apply CANT 8/8 to all trunks to a damaged switch. This control would cancel all the traffic destined for the damaged switch during the interim. If we try this CANT action with the simulator, we get the following results:

	GOS	MTFS	CFR
CANT 8/8 all trunks to TGS	0.045	1.055	0.0161

Here we should compare the 0.045 GOS with the 0.185 obtained with no controls. Clearly, the control action has improved GOS as seen by the controller, but 1.6% of the calls are failing altogether whereas none failed with no control action. The extra call failures are due to the cancellation of calls that are trying to use TGS as a tandem switch. We conclude that this control action is not indicated for the general case of switch damage. In the special case where the damaged switch has no tandem function the control would be beneficial, but the loss of such a switch has little impact on overall network performance in any event.

4.5.2 Switch Damage - PRL (Pearl Harbor, Hawaii)

PRL is the primary switch in the Hawaiian region. It is also the largest switch in the network, both in terms of trunk terminations and traffic. It has a significant role as a tandem switch for Western Pacific traffic to and from CONUS as well as in and out of the Hawaiian region. The loss of PRL therefore has significant impact on both Eastern and Western Pacific busy hour traffic, but since the EP traffic is more stressful, we present results for it here.

If we damage PRL and apply no NM controls, we observe the following:

	GOS	MTFS	CFR
No controls	0.657	1.869	0.4634

Under these conditions the apparent GOS which includes the failing attempts to reach PRL is 0.802. This is a severe damage situation. The controller sees 80% of call attempts being blocked, and the simulator shows that almost half the call intentions to destinations other than PRL are

failing to complete. Control action is clearly needed. Code Block (CB) for all calls to PRL yields:

	GOS	MTFS	CFR
CB 8/8 to PRL	0.692	2.468	0.1639

This control action has improved CFR significantly but worsened GOS and MTFS. This behavior appears to be typical of CB vs. no control in heavy traffic situations. Our analysis is that, without the code blocking, higher precedence calls destined for PRL preempt routine calls to other destinations but do not use the freed trunks because they cannot complete to the damaged switch. These trunks are then available for a succeeding attempt by a routine call as evidenced by the better grade of service. However, even though the routine attempt has a greater probability of success in the no-control case, it has a still greater probability of being preempted with the result that the chances for a satisfactory conclusion are less (indicated by a worse CFR).

It is reasonable to ask what would happen in this scenario if we put CANT controls on the trunks to PRL to help the situation while the CB controls were being installed. Such action would produce the following result:

	GOS	MTFS	CFR
CANT 8/8 to PRL	0.433	1.014	0.3049

These numbers appear to be better than the no-control results, but they fail to show that some precedence calls to destinations other than PRL, all of which succeed in the no-control case, are cancelled by the CANT controls. We feel that this property of the CANT makes it inappropriate for use in this situation.

With CB 8/8 to PRL in place, we can ask what further control actions might help the problem. Inspection of the point-to-point statistics from the simulator and the switch reports shows that most of the call failures and GOS problems are caused by traffic between the remaining switches in the Hawaiian region and the rest of the network. Examination of Fig 3.1 shows that this traffic must come and go through the SCS switch. Since the trunks out of SCS are fully loaded, there are no possibilities for using alternate routes to alleviate the problem at SCS. Cancellation of some of the excess traffic is indicated. Traffic into the Hawaiian region could be cancelled by Code Blocking or Call Gapping in combination with directionalization to avoid bias, but such a plan would be cumbersome to implement. A more direct approach would be to apply CANT controls to the trunks that are overflowing.

With the CANT control we have a choice of cancelling a fraction of the traffic offered to a trunk group or of cancelling calls that arrive faster than a threshold rate. The rate option is superior for use here since it will tend to become inactive if the traffic level drops below the level at which we set the threshold and will function more strongly if the traffic increases. In either case we need to observe the traffic before the control application, set the threshold, apply the control, observe the effect, and repeat the sequence until the desired result is obtained.

We carried out a series of experiments applying the rate flavor of CANT to the trunks between the remaining Hawaiian region switches and SCS. The results showed that the CFR could be improved a couple of percentage points beyond the CB-only results shown above by a number of different combinations of controls. The best result we obtained was the following:

	GOS	MTFS	CFR
CB 8/8 to PRL plus			
CANT > 600/min SCS <-> HIK, FSR			
CANT > 240/min			
SCS <-> CSH	0.527	2.010	0.1385

To examine the sensitivity of the control action to traffic level, we left the controls as just specified, reduced the traffic by 10%, and obtained the following result:

	GOS	MTFS	CFR
As above with 90% traffic	0.428	1.675	0.0946

This result should be compared with the case for the same traffic and only the CB to PRL controls in place. For that we got:

	GOS	MTFS	CFR
CB only with 90% traffic	0.553	1.899	0.0774

These results show that the CANT controls are cancelling more traffic than necessary for this rather small drop in traffic, and that consequently, if this type of control action is to be applied, it must be monitored closely and adjusted or removed when the traffic changes.

4.5.3 Gateway Switch Damage - LOD (Lodi, California)

The LOD switch is one of the two gateways between CONUS and DSN-PAC. Its failure results in loss of almost half the transmission resources between DSN-PAC and CONUS. To carry out the experiments with LOD damage, we used manually edited Eastern and Western Pacific busy hour traffic files which moved the LOD-sourced traffic to the other gateway (SLO) with the exception of traffic between LOD and SLO that was removed from the files. We do not know to what extent the resulting traffic patterns are realistic

for such a fault, because we do not know whether the CONUS-DSN would succeed in rerouting all the busy hour traffic through the other gateway.

With this scenario and no controls, we observe the following:

	GOS	MTFS	CFR
No controls	0.598	1.315	0.1269

For comparison, Eastern Pacific busy hour traffic with no damage gives:

	GOS	MTFS	CFR
Baseline, no damage	0.093	1.183	0.0005

Code blocking is not needed in this situation because no traffic terminates at LOD. The failing calls are all routine and are the result of overloading the remaining trunks between CONUS and DSN-PAC. There are no possibilities for routing modifications since the available trunks to and from SLO are fully loaded. Partial Code Blocking or Call Gapping could be used to reduce the overload, and our experiments show that if a 2/8 CB were applied in each direction (to CONUS destinations in the DSN-PAC and vice-versa in CONUS DSN) we would get the following improvement:

	GOS	MTFS	CFR
CB 2/8 each way	0.246	1.293	0.1170

Alternatively, we could apply CANT controls to the overloaded trunks. Since most of the traffic through SLO in the EP busy hour is between Hawaii and CONUS, we tried a fractional CANT control on the trunk groups between SLO and PRL and between SLO and SCS. The results were:

	GOS	MTFS	CFR
CANT 1/8 each way			
SLO <-> PRL			
SLO <-> SCS	0.367	1.392	0.1129

This control action gives a slightly better CFR and a little worse GOS, but it biases the failures against Hawaiian traffic in a way that the CB does not. However, it is much faster and easier to apply and remove when the traffic changes, and the bias it introduces is quite small. Extending the CANT to the other trunk groups out of SLO could in principle reduce the bias, but it is hard to avoid cancelling too much traffic because the quantization on the controls is rather coarse.

Neither control approach is needed for WP busy hour traffic which shows the following for LOD damage:

	GOS	MTFS	CFR
No controls	0.026	1.080	0.0000

4.5.4 Trunk Outage - YSM <-> OSK

In this scenario we examine the effect on network performance of the loss of a large intra-regional trunk group between YSM and OSK in Korea. Western Pacific busy hour traffic was used in the experiments. The results showed:

	GOS	MTFS	CFR
No controls	0.043	1.080	0.0013

CFR in this case is not worrisome, but it is worse than the zero value we get for the same traffic without damage. Examination of the switch reports shows that failing calls are all between YSM and OSK and all in the direction from YSM to OSK. Further, the reports show that trunk resources in the Korean region are not generally overloaded at this traffic level. Those facts suggest that control actions to modify routing behavior may be indicated. The routing table entry for YSM to OSK shows the first choice

to be the direct trunk that we have damaged in this scenario. The second choice is to go by way of TGS. The third uses YSS, and the fourth WLK. In all cases the route control digit requires that only the first choice route be used at the intermediate node. For TGS that first choice is the direct trunk group to OSK. Reference to Figure 3.1 shows that trunk group to be of size 15. It turns out that 15 trunks are not enough to carry the busy hour traffic between YSM and OSK. Other trunking is available, but the routing tables do not allow it to be used. There are two NM solutions to this problem. One is to modify the routing tables. The other is to put a fractional Skip control (SK) on the trunk group between YSM and TGS so that some of the traffic will try the third choice route even though the second choice trunk group is not fully occupied. We tried the fractional SK control and got the following result:

	GOS	MTFS	CFR
SK 5/8 YSM -> TGS	0.017	1.062	0.0000

This use of SK is not the ideal solution to the problem because it may cause a call that should go directly to TGS to take a longer path or perhaps to fail. If the SK applied only to traffic destined for OSK, this side effect would not be important, but SK applies equally to all traffic offered to the trunk group.

The modification needed to the routing table at YSM would be to change the control digit to two so that the second choice route at TGS could be used.

Two other trunk outage scenarios have been run. One damaged the small trunk group (7 trunks) between CKS and SLO. It showed a minor increase in

GOS but no calls failed. The other involved partial loss (20%) of the large group (120 trunks) between PRL and LOD. The effect was relatively greater since the group is heavily loaded under EP busy hour conditions. In this scenario we were not able to find any control action that would improve GOS without worsening CFR from its no-control value of 0.0070.

4.5.5 Two-Switch Damage Scenarios

As stated in Section 3.3, we have explored three two-switch damage scenarios. There are not enough items of interest in the results to warrant their presentation here in detail. The behavior was very similar to that of the single-switch cases. Overall GOS and CFR were worse, but not dramatically so, and Code Block to the damaged switches gave the same kind of improvement. The main point of note is that failure of higher precedence calls was observed in two out of the three scenarios. The failures went all the way to the FLASH level in one scenario. The problem was traced to the routing tables, in particular to the route control digit. When the control digit limits the choice to the first n routes, and the corresponding switches are damaged, the call will fail independent of its precedence level. In the routing tables there are places where the control digit allows only the first two routes to be used for some source/destination pairs, and the corresponding two nodes happened to be damaged in our scenarios. Study of the tables showed that it was possible to change the control digit to three without introducing routing loops and thereby rescue all the precedence traffic. In the course of the investigation, we also observed that the May 1987 set of routing tables had the value of three in many places where the more recent tables have the

value of two. We changed to the older tables in one scenario and found that overall performance was slightly better and that there were no failures of precedence calls. We do not know why the values were reduced to two in the more recent tables.

In the scenario in which we damaged both YOK and SCS, we observed that with WP traffic the trunks going west from PRL were overloaded. At the same time the trunks to LOD and SLO were lightly loaded and that there was some spare capacity from those switches to the west. We therefore tried an experimental routing table that allowed those resources to be used for traffic between Hawaii and the western Pacific. The modified table improved CFR from 0.0928 to 0.0821 for WP traffic. The change would not be appropriate for EP traffic since the resources would not be spare with that traffic pattern.

4.6 Conclusions and Observations

4.6.1 General Conclusions

The principal conclusion from our NM experiments is that appropriate NM control actions are critically dependent upon the traffic patterns of the moment and only secondarily dependent upon the particulars of a damage situation. The locations at which controls might be applied obviously depend upon the damage, but the choice of controls, their parameter settings, and whether or not any control is needed at all depend directly on the observed traffic. We believe this conclusion results from the inherently robust design of the network which provides good resistance to damage. Performance degrades gracefully and slowly as long as the users are willing to retry their calls with persistence.

A second conclusion is that 100% Code Blocking to unreachable destinations is always a good NM action, primarily because it excludes from the network precedence traffic that cannot succeed. Such traffic, if allowed to enter the net, will preempt calls that would otherwise be successful. The CB action is also beneficial in reducing the time that trunk and switching resources waste in attempting to handle calls that cannot succeed. This benefit is particularly important when the traffic offered to unreachable destinations is heavy. A third significant benefit is that Code Blocking allows the controller to more clearly observe what is happening to traffic that has a chance to succeed by removing from the switch reports the statistics for the calls attempts that have no chance. Because of retrying, the latter traffic may well dominate the switch report statistics.

A third conclusion that can be drawn from our experiments is that there are some traffic congestion situations that can be alleviated by modifying the routing behavior of the network. Such modification can sometimes be accomplished by the use of Skip controls, but in other cases changes to the routing tables may be needed. We think of these as "expansive" NM actions that allow the use of otherwise wasted network resources. Like all traffic-dependent actions, they should be removed or adjusted as the traffic changes.

A fourth conclusion is that damage scenarios will often be accompanied by traffic patterns in which the offered traffic will exceed the capacity of the network transmission resources after all expansive NM possibilities have been exhausted, and call failures will be unavoidable. In such cases,

it is often possible to make a small improvement in the CFR by judicious use of restrictive controls such as CANT, partial Code Blocking or Call Gapping. Beyond that point, it may be the case that GOS and MTFS can be improved without significantly worsening CFR. The benefits from these actions depend upon the notion that a cancelled call is not retried. Such is the case in CCSIM, but in the real world retrying depends upon the behavior of the user. To get the effect seen in the simulations, it is important for the user to get a message that causes him to not retry his call when his attempt is cancelled by NM control action.

A fifth conclusion from our experiments is that restrictive NM controls are very sensitive to traffic levels and patterns, and that their application requires careful monitoring to avoid degrading network performance beyond that caused by the damage alone. Controller expertise and good displays of network status information are clearly needed. We do not yet have either and can make no claims that the experimental results cited above (Section 4.5) represent any sort of optimal performance for the control actions taken.

4.6.2 Observations on Individual NM Controls

4.6.2.1 Code Block (CB) vs. Call Gap (GAP)

Partial CB is easier to apply than GAP because it requires less knowledge about the offered traffic pattern. For example, if we want to reduce the traffic from DSN-PAC to CONUS, we can set a fractional CB at each switch, and reduce the traffic in a nondiscriminatory way. To use GAP we would need to select an appropriate gap threshold for each switch's traffic to CONUS, and to do so without bias, we would need some information

about each switch's requirements and/or normal traffic to CONUS at the particular time of day that the control was to be introduced.

4.6.2.2 CANT

In CCSIM we have two flavors of the CANT control. One cancels a fraction of the traffic offered to a trunk group. The other applies a call gapping gate to calls offered to the group. This flavor closes a gate for a time after any call passes through the gate and cancels all calls that arrive while the gate is closed. The latter action is better suited to handling overloaded trunk groups because it will tend to become inactive if the traffic level drops and will operate to cancel more effectively if the traffic increases above the level at which the gate was set. Unfortunately, statistical variability of call arrivals is such that the gate will cancel some calls even though the average rate is substantially below the nominal threshold.

Both flavors of CANT have rather coarse parameter settings which make them somewhat difficult to use for fine tuning purposes. In that regard, we have observed that the best compromise setting for the CANT control is one that leaves, in any switch reporting interval, several times as many calls overflowing the trunk group as are cancelled. If enough calls are cancelled so that the group does not overflow, CFR will be unnecessarily worsened.

4.6.2.3 Skip (SK)

We use the SK control for two purposes. One is to tweak the routing mechanism as illustrated in the scenario described in Section 4.5.4. The other is to cause a switch to avoid using a trunk to a damaged switch. In

our simulation, the neighbors of a damaged switch are not made aware of the damage when it occurs. They discover that the individual trunks to the damaged switch are not usable and busy them out one at a time as they try to use them for routing calls. If the traffic level is low, it can take some time for all the trunks in the group to be busied out. In the meantime, calls that might attempt to use the damaged switch as a tandem switch may be blocked unnecessarily. Application of the SK control to such a group will avoid such guaranteed blockage by trying any allowable alternates. However, we have observed that if the SK control is applied before an appropriate Code Block takes effect, the SK will magnify the damage caused by unsuccessful precedence calls by extending their paths as they attempt to reach the damaged switch. Thus we tend to get better performance if SK is applied in conjunction with CB and worse if the SK is applied to such trunks alone.

4.7 Recommendations

4.7.1 NM Control Actions

Based on experience to date, our recommendations for NM control actions to be taken in the event of damage in DSN-PAC are as follows:

(1) For switch damage other than gateways, apply Code Block (CB) to destinations reachable only through the damaged switches. (For gateway damage, start with (2). For trunk outages start with (3).) While waiting for these controls to take effect, do nothing.

(2) When the CB controls (if any) are in place, put 100% SK controls on all the trunk groups to the damaged switches.

(3) Examine switch reports for evidence of excessive blocking. Overflows from trunk groups are to be expected in damage scenarios

and heavy traffic situations. They are not worrisome unless there is evidence of excessive blocking. If none is found, there is no need for further control action, but the situation should be carefully watched so that if traffic increases, action can be taken. If excessive blocking is present in parts of the net, (4) and/or (5) may be applicable.

(4) If the switch reports show evidence of heavy overload, it is unlikely that significant help can be gained by attempting to use trunk resources that are not fully utilized. In such a case, proceed to (5). If the reports show only moderate overloading with the traffic of the moment, and some resources are not being fully utilized while others are saturated, explore the possibilities for using Skip (SK) controls or routing table changes to make use of the spare trunk resources. These actions should only be undertaken by experienced personnel who understand the operation of the routing algorithm in the network. Serious damage to network performance could result from inappropriate routing table changes.

(5) If heavy overload conditions exist, or moderate overloads remain after any expansive controls under (4) have been implemented, the use of restrictive controls such as CANT should be explored. (We expect that Alternate Route Cancellation (ARC) will also be applicable, but we have not yet explored its use in any experiments.) CANT should be applied to overflowing trunk groups that are the last groups in routing chains. In CCSIM switch reports, we show such end-of-chain overflows explicitly. If they

are not available in the real network's switch reports, knowledge of routing tables together with assumptions about the traffic can be used to infer which overflows are likely to correspond to end-of-chain. The number of outgoing incomplete call attempts should correlate with end-of-chain overflows. We recommend use of the gapping flavor of CANT with the parameter set so that the overflows remaining after the control is in place will be larger than the number of calls cancelled by the control. Simulation results show that cancelling enough traffic to bring the overflows to zero will result in excessive call failures. The idea is to cancel as little traffic as necessary. Unfortunately, we do not yet have a formula relating observed GOS and/or trunk group overflows to CFR which would allow a precise recommendation for parameter setting to be made. Our experiments show that the best results are obtained with a ratio of 2 to 4 between the overflows and the number of calls cancelled. The recommended procedure is to pick a setting, observe the result, and adjust as required to keep the cancelled calls within the suggested range. That range depends upon the retrying behavior of the users. On the average, a cancelled call removes n call attempts from the network, where n depends upon the average retry persistence of the users.

Our experiments show that if the expertise to follow procedures (4) and (5) is not available, or the switch reports are not coming through to the controller, the consequences of omitting those actions are not severe. The benefits to be expected from these procedures are typically the saving

of a few percent of the call intentions. The penalties for misuse are likely to be an equal or greater increase in the failure rate.

4.7.2 General Recommendations

(1) If NM controls that cancel calls (CB, GAP, and CANT) are to accomplish their purpose, it is important that callers receive a message that effectively prevents retry of the call in the event that a call is affected by the control. A message or signal interpreted by the caller as equivalent to a blocked call indication will result in the control action having no useful effect.

(2) Retrying persistently is important for achieving a useful CFR in a damaged network. Therefore, we would recommend against administrative actions to limit retries for damage scenarios no more severe than the two-switch busy hour ones we have investigated. On the other hand, administrative action to discourage unnecessary use will always help.

5. PROTOTYPE NETWORK MANAGEMENT EXPERT SYSTEM

5.1 Current Status

This section provides a brief summary of the current status of the prototype Expert System implemented in FY87. A more detailed description is given in Appendix B.

The system concept we are addressing is that of two SUN work stations coupled by an Ethernet, with one of them hosting CCSIM and the other hosting the Expert System. Switch reports flow from CCSIM to the Expert System, and network management control commands flow in the other direction. As described in Appendix B, this model represents the future deployed Expert System interacting with the real DSN.

A flexible, powerful development environment has been created for the Expert System. It consists of a SUN work station similar to that supporting CCSIM, except that it has twice the memory (16 megabytes, compared to 8) in order to minimize disk swapping. SUN Common LISP is loaded in the system, on top of its native UNIX operating system; and the ART expert system development shell (a product of Inference Corporation) runs on top of LISP. SUNview, a sophisticated C-language graphics package, provides identical graphics facilities for both CCSIM and the expert system.

A functional architecture has been developed for the Expert System in which many simple, specialized "monitors" continually scan incoming switch statistics reports for patterns and features of interest. A higher-level abstract model of the network is maintained within the system, consisting only of the interesting or unusual facts that are currently in evidence. This is a very small body of information, compared to the enormous size of the fact base that would be needed to give a complete description of the network and its traffic, and this winnowing process exemplifies the power of expert systems in extracting the important essence from large bodies of information. Another module in the expert system continually reviews the network state, seeking evidence that a problem situation is arising. Whenever such evidence is discovered, a confirmation module pays close attention to that factor to ascertain whether it persists over time. If so, a planning module is called upon to devise a set of control actions to suppress or correct the problem. Meanwhile, the monitors look for evidence in subsequent network behavior that the control actions are in fact having the desired effects. Any of these modules has the capability to activate a

suitably tailored set of monitors (of which there could be dozens, or hundreds) to observe a particular effect it is looking for.

At present a number of elements of the expert system have been implemented. These include several different monitors, which can be watched in action as they review switch statistics reports (from pre-stored files) and identify features. Facilities have been provided for rapidly translating conventional network connectivity and traffic files into forms that can be used internally by the Expert System. Designs are in hand for certain higher-level modules, and others are in the design process. It is expected that the FY88 efforts at Lincoln Laboratory under this program will lead to a fully integrated system as described above, with the Expert System conducting interactive network management of a simulated network running in CCSIM.

5.2 Assessment of Future Potential

As noted above, a design has been laid out for the DSN Network Management Expert System which appears capable of embracing the general problem of managing the simulated DSN running within CCSIM. Some portions of this design have been implemented and tested, such as examples of monitors and the facilities for transforming network representations and switch reports into ART-compatible structures. Much remains to be done, and an honest appraisal is that some aspects of the system (such as the abstract network state representation, and the planning/replanning module) pose challenging problems.

On the other hand, it is clear that we have constructed a flexible and powerful development environment. It is also clear that a series of straightforward intermediate steps can be defined as a graded approach

toward the sophisticated system required for full satisfaction of the overall objectives.

We have earlier estimated that, starting from our state of knowledge and development as of the end of FY87, about four person-years would be required to produce an Integrated DSN Simulation (IDSIM) at Lincoln Laboratory. This integrated system would consist of CCSIM and the Network Management Expert System, each running in its own SUN work station as indicated in Fig. B.1, with switch reports and network control commands flowing between them on an Ethernet. The expert system would incorporate the knowledge and facilities necessary to successfully manage a DSN simulation in CCSIM through a wide-ranging variety of realistic congestion and damage scenarios. Efforts would be made to correlate the IDSIM graphics and user interface facilities with those developed under DCOSS, and provisions would be made for applying IDSIM as an aid in training DCOSS network management personnel. We see no reason to change this four person-year estimate at this time. Crudely speaking, for a 1-year program the estimate assumes one person doing expert system design and implementation; two and one-half people doing NM knowledge engineering; and one person half-time doing CCSIM modifications and extensions.

Estimating the magnitude and complexity of the job of transforming the IDSIM engineering tool into an operational NM expert system for deployment in the DCS, on the other hand, is a more difficult task. Instead of a laboratory model which must be maintained by skilled software engineers, the deployment system would have to be capable of withstanding any physical or operational abuse it would be likely to receive. Its catalog of skills would have to have been fully tested under all

appropriate variations of network performance. If it were not thoroughly dependable, the field personnel would quickly conclude that it was too risky to use. An organization capable of creating this operational system would need to be thoroughly experienced in specifying, creating, validating and testing military software systems. Starting with IDSIM as it is expected to exist at the end of FY88, it is probably optimistic to expect that the job could be completed with less than five additional person-years of effort. To this must be added the costs of hardware and software facilities to host each deployed expert system; for reference purposes, the cost of a SUN work station similar to the IDSIM equipment is in the range of $60,000 (including a hard disk and software licenses).

APPENDIX A

GUIDE TO CCSIM OPERATION, OCTOBER 1987

The purpose of this Appendix is to provide a concise set of instructions, illustrated by example, for operation of the Call-by-Call Simulator (CCSIM) as of the end of October 1987. It is assumed that the user environment is a SUN 3/260 work station essentially identical to those at Lincoln, and that it has been provided with all the files and facilities described below.

A-1. Functional Overview

The documentation delivered to DCEC with the CCSIM at the end of FY85 included a "User's Guide for the Lincoln Laboratory Circuit-Switch Network Call-by-Call Simulator" by R. P. Lippmann and Robin Krumholtz, which continues to be valid in a general sense. The current CCSIM is different in a number of specific ways, however, with respect to functional capability (as described in the body of the FY87 Annual Report) as well as operating procedures, as detailed in this Appendix.

The FY85 CCSIM was normally operated in a batch mode, with all the commands and parameters for the run supplied in an INVAL file. The results of the batch run were chosen from a variety of table and plot options. From an operations point of view, the primary change in the FY87 CCSIM is the notion of interacting with the system in real time, with successive network damage events, control actions, and recovery cycles. To support this mode of operation a graphics display system is in the process of implementation, including a network topological map, graphical indications of various parameters and conditions, and a mouse-sensitive menu of commands to issue to CCSIM. Ultimately the Network Management Expert

System, with real-time switch statistics inputs from CCSIM, will be available to aid the human operator in evaluating conditions in CCSIM and choosing appropriate control actions.

For the present, a command file mode of operation has been implemented. A sequence of damage events and control actions is made up in advance, each tagged with the time it goes into effect, and stored in a file having the suffix .cmds in its name which is read by CCSIM as it runs. The effect is equivalent to a user typing in the commands at the associated times. By altering the command file for successive runs, the user can conveniently run coherent series of experiments exploring the effects of particular control strategies, all using the same set of network and traffic conditions (specified in traffic and INVAL files, essentially as in FY85). This process will be illustrated in examples to follow.

Since there are now three possible sources of commands to CCSIM, new INVAL parameters have been added to specify which source is active for the current run. The parameter names are grphic , expert , and comman , and their present default values are 0, 0 and 1, respectively.

Another significant operation-related change in the FY87 CCSIM is the addition of four new matrix formats for call statistics, giving much more detail than was formerly available. For every individual source-destination pair possible in the network, the four matrices list the blocking probability, the number of offered calls, the number of blocked calls, and the number of calls affected by code cancellation or gapping controls during the run. Moreover, this set of four matrices is presented separately for each of the five precedence levels. This information gives a

very detailed and complete picture of what is going on in the net, and is valuable for many purposes, but is far more than will ever be known to DSN network managers. Their information will be received in the form of a statistics report received once in each five-minute interval from each switch in the network. Accordingly CCSIM has been modified to produce these reports, as described in the body of the FY87 annual report. Examples are given below.

A-2. CCSIM Operating Environment

A set of convenient files and "C" shell scripts has been created at Lincoln to simplify the execution of CCSIM runs. When the SUN work stations are delivered to DCEC, Lincoln personnel will set up and check out similar facilities there.

The following paragraphs list and describe most of the important files for a user named "hmh". (Other users can have their files in nearby directories if they wish, distinguished by other prefixes.)

The input information for a particular run is contained in the traffic and INVAL files hmh.traf and hmh.inval, and in the command file hmh.cmds. The first two contain baseline network, traffic and simulation parameter information, and are similar to files used in setting up the FY85 CCSIM. The hmh.cmds file is edited to suit the current run. A convenient template for the hmh.cmds file is provided by reading in a standard command file std.cmds stored in the user's directory, and using the editor to replicate and modify lines for the current run. Since the Fortran code of CCSIM requires precise input formats, this replication scheme helps to avoid mistakes. The contents of the std.cmds file are as follows:

```
destroy-node        PRL
destroy-trunks      CKS     FBK     SAT     14
directionalize      CKS     FBK     LAND    7
time                3500
switch-report-on    SCS
cancel-to-link      CKS     FBK     SAT     8
cancel-alternates   SCS
skip-link           SCS     PRL     LAND    7
time                4000
switch-report-off   SCS
code-cancel         ALL     PRL     4
time                4000
restore-node        PRL
restore-trunks      CKS     FBK     SAT     14
code-gap            ALL     FBK     15
gap-to-link         CKS     FBK     SAT     10 end
```

Section A-3 of this Appendix explains in detail how to create a command file to runm CCSIM, including full descriptions of all available commands and their arguments. Briefly, the first line of the std.cmds file specifies the time (in 0.1-second ticks since the start of the run) at which the succeeding list of one or more commands, up to the next time line, are to take effect. The "destroy" and "restore" commands operate on the indicated nodes and trunks. The network controls available are directionalize, cancel-to-link, cancel-alternates, skip-link, code-cancel, code-gap, and gap-to-link, each with a suitable set of arguments. The switch report on and off commands allow the user to collect switch statistics information from only the switches of current interest, during only the time periods when interesting behavior is likely (if left on continually, the switch reports would produce enormous volumes of uninteresting data).

A file named runsim is an executable shell script which carries out all the steps of initializing and running CCSIM, and collecting and

printing out the resulting data. It is invoked by typing

runsim "comment field"

with the comment field containing a few words that will help the user remember the conditions and goals of the run. The primary output statistics are stored in a succession of files named hmh.run1, hmh.run2, ... , in which the ending digit is automatically incremented each time a new run is executed. If switch reports were turned on during the run, they are saved in a file named (e.g.) hmh.sw.run3 in a verbose, labeled form that is easily interpreted by a human operator. The same switch reports are saved in a file named (e.g.) hmh.exp.run3, in the very compact unlabeled format in which they would be transmitted to the expert system. To assist the user in remembering and interpreting his work, a file named runlog is automatically augmented by runsim each time a new run is executed, listing the date, time, and a brief description of the run.

A-3. Definitions of Command File Functions

The Command file is a formatted simulator input file with the suffix .cmds. To read the command file there must be a line in the .inval simulator input file setting comman equal to one (comman=1;). The following controls can be issued from the command file to the simulator:

Network Management Controls:

1. ARC
2. CANT
3. CB
4. DRZ
5. GAP
6. SK

Simulator Controls:

1. destroy-node
2. destroy-trunks
3. restore-node
4. restore-trunks
5. sw-report-off
6. sw-report-on
7. vsw-report-off
8. vsw-report-on

Command File Controls:

1. end
2. time

The commands in the file must be in the correct format to be read in to the simulator. This format is as follows:

column 1 - the command name must start in column 1
column 20 - the first variable must start in column 20
column 28 - the second variable must start in column 28
column 36 - the third variable must start in column 36
column 44 - the fourth variable must start in column 44

Command Descriptions:

Network Management Controls:

ARC

Alternate Route Cancellation.

Cancel alternate routes out of a node or all nodes.

Usage: ARC node

where node is a three letter node name or the word ALL

Example: ARC SCS

Cancels the alternate routes out of SCS.

CANT

Cancel To.

CANT-RATE

Set an upper limit on the rate at which calls are allowed to access a trunk group.

Usage: CANT-RATE node1 node2 gapping-index

where node1 is a three letter node name or the word ALL

node2 is a three letter node name

gapping-index is an index value into a table containing gap intervals.

Gapping Index	Gap Interval (Seconds/Call)	Calls per Minute
1	0	All Calls allowed
2	0.1	600
3	0.2	240
4	0.50	120
5	1	60
6	2	30
7	5	12
8	10	6
9	15	4
10	30	2
11	60	1
12	120	1/2
13	300	1/5
14	600	1/10
15	Infinity	All Calls blocked

Example: CANT-RATE CKS FBK 10

Allows 2 calls through per minute on the trunk groups from CKS to FBK.

CANT-%

Cancel a fraction (in eighths) of attempts to access trunk groups from one node to another.

Usage: CANT-% node1 node2 percent

where node1 is a three letter node name or the word ALL

node2 is a three letter node name

percent is the percentage in eighths(0-8)

Example: CANT-% CKS FBK 8

Cancels 100 percent of the calls on the trunk groups from CKS to FBK.

CB

Code Block

Cancel a fraction (in eighths) of calls from one node to another node.

Usage: CB node1 node2 percent

where node1 is a three letter node name or the word ALL

node2 is a three letter node name

percent is the percentage in eighths(0-8)

Example: CB ALL PRL 4

Cancels 50 percent(4/8) of the calls from all nodes to PRL.

DRZ

Directionalization.

Directionalize a number of trunks on a link. Limits the number of trunks a node can use to send calls out on a link.

Usage: DRZ node1 node2 type number

where node1 is a three letter node name or the word ALL

node2 is a three letter node name

type is either LAND, SAT, or ALL

number is the number of trunks on link a node can use to send calls on.

Example: DRZ CKS FBK LAND 7

Allows CKS to use 7 trunks on its land link to FBK to send calls out.

GAP

Call Gapping

Set an upper limit on the rate at which attempts to a particular code are allowed. Cancels all calls over the rate.

Usage: GAP node1 node2 gapping-index

where node1 is a three letter node name or the word ALL

node2 is a three letter node name

gapping-index is an index value into a table containing gap intervals.

Gapping Index	Gap Interval (Seconds/Call)	Calls per Minute
1	0	All Calls allowed
2	0.1	600
3	0.2	240
4	0.50	120
5	1	60
6	2	30
7	5	12
8	10	6
9	15	4
10	30	2
11	60	1
12	120	1/2
13	300	1/5
14	600	1/10
15	Infinity	All Calls blocked

Example: GAP ALL FBK 15

Allows no calls from all nodes to FBK through.

SK

Skip.

Skip a specified link a percentage of the times it would be accessed.

Usage: SK node1 node2 type percent

where node1 is a three letter node name or the word ALL

node2 is a three letter node name

type is either LAND, SAT, or ALL

percent is the percentage in eighths(0-8)

Example: SK SCS PRL LAND 7

Skips 7/8 of the calls on the land link from SCS to PRL.

Simulator Controls:

destroy-node

Destroy one node or all nodes.

Usage: destroy-node node

where node is a three letter node name or the word ALL

Example: destroy-node PRL

Destroys PRL.

destroy-trunks

Destroy a specified number of trunks on a link.

Usage: destroy-trunks node1 node2 type number

where node1 is a three letter node name or the word ALL

node2 is a three letter node name

type is either LAND, SAT, or ALL

number is the number of trunks on link to destroy

Example: destroy-trunks CKS FBK SAT 14

Destroys 14 trunks on the satillite link from CKS to FBK.

restore-node

Restore one node or all nodes.

Usage: restore-node node

where node is a three letter node name or the word ALL

Example: restore-node PRL

Restores PRL.

restore-trunks

Restore a specified number of trunks on a link.

Usage: restore-trunks node1 node2 type number

where node1 is a three letter node name or the word ALL

node2 is a three letter node name

type is either LAND, SAT, or ALL

number is the number of trunks on link to restore

Example: restore-trunks CKS FBK SAT 14

Restores 14 trunks on the satillite link from CKS to FBK.

sw-report-off

Turn the switch reports off for one node or all nodes.

Usage: switch-report-off node

where node is a three letter node name or the word ALL

Example: switch-report-off SCS

Stops sending switch reports to the switch report file for SCS.

sw-report-on

Turn the switch reports on for one node or all nodes.

Usage: switch-report-on node

where node is a three letter node name or the word ALL

Example: switch-report-on SCS

Sends switch reports to the switch report file for SCS.

vsw-report-off

Turn the verbose switch reports off for one node or all nodes.

Usage: switch-report-off node

where node is a three letter node name or the word ALL

Example: switch-report-off SCS

Stops sending verbose switch reports to the verbose switch report file for SCS.

vsw-report-on

Turn the verbose switch reports on for one node or all nodes.

Usage: switch-report-on node

where node is a three letter node name or the word ALL

Example: switch-report-on SCS

Sends verbose switch reports to the verbose switch report file for SCS.

Command File Controls:

end

Indicates that no more commands will be read from the command file.

Usage: end

Example: end

No more commands are read from the command file after an end is read.

time

Indicates the time that the next command will occur.

Usage: time time

where time is the time in ticks (10ths of a second).

Example: time 3000

Sets the next command time to 3000 ticks.

NOTE: All node names and trunk types must be in capital letters

To read commands from a command file a new parameter must be set in the inval file and the parameter indicating that graphics will be used must be off:

comman=1,

grphic=0,

NOTE: Nodes and links can still be destroyed from the .inval file in the same way they were in the past.

The following is a sample .inval file.

```
 &inval
 comman=1,
 graphic=0,
 dtint=3000,
 dtplot=3000,
 lout=0,
 tdam = 0,
 gcall=0,
 newdet=2,
 nnod=23,
 perlod=1.00,
 swdel=100,
 lnkdel=600,
 pcbusy=.4,
 prbusy=.9,
 tmbusy=3000,
 trbusy=600,
 reblok=9,
 prblok=.90,
 trblok=600,
 redam=1,
 prdam=.90,
 trdam=600,
 reprem=9,
 prprem=.90,
 trprem=600,
 seed=12345,
 tstart=0,
 tvalid=0,
 tstop=12000,
 &end
 PACIFIC
 Resized simulator  Hldtim =time+(nn*swdel+(nn-1)*lnkdel)/200
 1
 8
 3
 1 8 0 6
 1 8 1 12
 2 8 0 3
```

```
Oct 28 09:17 1987  hmh.run4 Page 1

Run4 - Tango sw destroyed, code-cancel ALL to Tango

Wed Oct 28 09:12:50 PST 1987
-rwxrwxr-x  1 lynn       425984 Oct 27 12:44 /usr/romulus/ccsim/ccsim/src/sim/simulator*
-rw-r--r--  2 hal          3197 Oct 27 19:59 hmh.traf
-rw-r--r--  1 hal          3197 Oct 27 19:58 hmhep.traf
-rw-r--r--  2 hal          3197 Oct 27 19:59 hmhwp.traf
Wed Oct 28 09:16:15 PST 1987

Command file contents

time                36000
destroy-node        TGS
code-cancel         ALL     TGS     8
time                72000
switch-report-on    CRC
switch-report-on    YSM
end
```

Figure A.1.

A-4. Execution of an Example Run

As an illustration of CCSIM operation, we describe hmh.run4, in which the Tango switch in Korea was destroyed one hour into the run, using West-Pac busy-hour traffic conditions. At the same time the code-cancel control was implemented in all switches in the network, on 100% of the calls directed to Tango. The figures to follow are a few example pages of statistics printouts from this run. (For more detailed descriptions of the contents of the statistics reports, the reader is referred to the main body of the FY87 Annual Report.)

Figure A.1 is the first page of hmh.run4; it is headed by the user-defined comment field created when runsim was invoked. The list of UNIX files begins with the pathname of the simulator code that was actually executed. The second file (hmh.traf) is the traffic file accessed by the shell script runsim; prior to the run hmh.traf was linked to the source file hmhwp.traf representing West-Pac busy-hour traffic. The linking is indicated by the 2 in the 13th column of the listing of the two files. The user could instead have linked hmh.traf to the East-Pac busy-hour traffic file hmhep.traf, or to some other file as desired.

Figure A.1 also displays the contents of the command file used for the run. One hour (that is, 36,000 tenth-second ticks) into the run, the Tango node was destroyed and the "code-cancel" control was applied in ALL network nodes for calls going to TGS (i.e., Tango switch), effective for 8 eighths of such calls. One hour after the damage, to allow time for the statistics to settle down, switch reports were turned on for two of the switches having trunk connections to Tango, namely Clark Air Force Base (CKS) and Yongson-Main (YSM).

CALLS TRYING 1,2,3,... TIMES (OFFERED AFTER TVALID)

# CALLS =	8374	1028	608	433	356	293	254	220	193	164
GOS =	0.073	0.451	0.604	0.716	0.750	0.795	0.760	0.786	0.720	0.817
BLOCKED =	615	464	367	310	267	233	193	173	139	134
PEEEMPTED =	387	133	63	45	19	21	22	21	26	13
DROPPED =	0	0	0	0	0	0	0	0	0	147

CALLS of PRIORITY 1 ONLY

# CALLS =	5576	1014	599	429	352	290	253	219	193	164
GOS =	0.108	0.449	0.608	0.713	0.750	0.800	0.759	0.790	0.720	0.817
BLOCKED =	603	455	364	306	264	232	192	173	139	134
PEEEMPTED =	385	133	63	45	19	21	22	21	26	13
DROPPED =	0	0	0	0	0	0	0	0	0	147

CALLS of PRIORITY 2 ONLY

# CALLS =	2138	12	8	4	4	3	1	1	0	0
GOS =	0.005	0.667	0.375	1.000	0.750	0.333	1.000	0.000	0.000	0.000
BLOCKED =	10	8	3	4	3	1	1	0	0	0
PEEEMPTED =	2	0	0	0	0	0	0	0	0	0
DROPPED =	0	0	0	0	0	0	0	0	0	0

CALLS of PRIORITY 3 ONLY

# CALLS =	622	2	1	0	0	0	0	0	0	0
GOS =	0.003	0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
BLOCKED =	2	1	0	0	0	0	0	0	0	0
PEEEMPTED =	0	0	0	0	0	0	0	0	0	0
DROPPED =	0	0	0	0	0	0	0	0	0	0

CALLS of PRIORITY 4 ONLY

# CALLS =	32	0	0	0	0	0	0	0	0	0
GOS =	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
BLOCKED =	0	0	0	0	0	0	0	0	0	0
PEEEMPTED =	0	0	0	0	0	0	0	0	0	0
DROPPED =	0	0	0	0	0	0	0	0	0	0

CALLS of PRIORITY 5 ONLY

# CALLS =	6	0	0	0	0	0	0	0	0	0
GOS =	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
BLOCKED =	0	0	0	0	0	0	0	0	0	0
PEEEMPTED =	0	0	0	0	0	0	0	0	0	0
DROPPED =	0	0	0	0	0	0	0	0	0	0

Following are totals from Time Zero NOT Time Valid!

drop(tries)=	0	0	0	0	0	0	0	0	0	330

TOTAL OFFERED CALLS = 35660
TOTAL ORIGINAL CALLS = 26577
NOT OFFERED Damaged Source = 308
NOT OFFERED Code Cancelled = 432
NOT OFFERED BAD PARAM = 1
TOTAL DROPPED DEST-BUSY CALLS = 0
TOTAL NEW TRYS DEST-BUSY CALLS = 0

Figure A.2.

The second and third pages of hmh.run4 (not shown) give a summary description of the network and list the values of all the parameters in the INVAL file. Figure A.2 (page 4 of the listing) shows call success and retry statistics. The first block is the overall results for all call precedence levels, and the next five blocks treat the five precedence levels separately.

Figure A.3 is the matrix of call blocking performance for calls of precedence level 1 (or, equivalently, local grade of service) for all possible source/destination pairs among the 23 nodes in the network. The leading decimal points have been left off to simplify the format of the chart; that is, the numbers represent parts per thousand.

Figure A.4 lists the actual number of precedence-1 calls that were blocked for each source/destination pair. Figure A.5 is the matrix of offered precedence-1 calls, and Fig. A.6 lists all the calls that were affected by code cancellation.

Figure A.7 is the first page of the switch report file, hmh.sw.run4. At the top is the report from CRC (Camp Red Cloud, Korea) for the five minute interval beginning two hours into the run. First it lists the total numbers of incoming and outgoing calls, etc., for the entire node. The five columns of numbers refer to the five levels of call precedence. This is followed by a separate report for each trunk group connecting CRC with some other node. As it turns out, CRC has only two trunk groups to the rest of the world: one to YSM (Yongson-Main), and one to the Tango switch. Notice that all calls to TGS were overflowed (i.e., blocked), and the added stress on the only other set of outgoing trunks caused performance on it to be

BLOCKING MATRIX FOR PRECEDENCE LEVEL 1
FROM 2: 0: 0 TO 3: 0: 0 DURATION 1: 0: 0 = 36001 TICKS

Source	Destination ----------> 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	NR Offered by Source
1/CKS	0	0	0	235	0	307	0	62	26	166	0	0	0	0	55	15	0	0	26	0	166	0	950	494
2/FGB	35	0	0	250	0	208	166	295	0	0	0	90	0	0	0	100	333	0	0	0	0	0	500	191
3/FBK	55	43	0	228	0	396	363	66	125	66	0	0	0	0	125	0	0	0	0	0	0	0	823	525
4/YOK	274	222	169	0	17	158	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	666	504
5/CZA	500	0	361	68	0	199	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	933	343
6/YSU	264	444	329	55	41	0	117	45	83	125	0	0	0	0	0	42	121	500	95	0	399	0	814	480
7/IWK	428	0	399	0	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75
8/PRL	78	214	250	0	0	130	0	0	0	0	0	0	0	0	0	0	125	0	0	0	0	0	750	281
9/SCS	43	199	303	0	0	111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	857	296
10/FSR	352	500	111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166	0	0	0	500	123
11/HIK	0	199	300	0	0	399	0	0	0	0	0	0	0	0	0	0	0	0	375	0	250	0	0	252
12/CSH	266	0	125	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	333	0	0	0	0	163
13/LOD	149	0	285	0	0	79	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	769	156
14/SLO	161	399	90	0	0	68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	571	152
15/DRA	0	500	333	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22
16/CON	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17/YSM	125	444	100	0	19	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	399	785	1411
18/YSS	0	0	0	0	0	0	500	199	0	0	0	0	0	0	0	71	0	0	0	0	0	272	780	926
19/OSK	13	0	0	0	0	0	0	0	66	0	0	0	0	0	0	0	0	0	0	0	0	0	771	480
20/TGS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21/WLK	0	0	0	0	0	500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	802	472
22/PSK	142	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	775	103
23/CRC	952	0	827	740	923	814	0	695	900	0	0	999	0	0	0	732	794	753	820	0	806	818	0	1640
Total Offered To Destination	475	120	425	342	527	563	113	362	364	219	170	159	2	2	42	405	1478	520	668	0	544	137	1452	9089 = 9089

Figure A.3.

BLOCKED CALL MATRIX FOR PRECEDENCE LEVEL 1
FROM 2: 0: 0 TO 3: 0: 0 DURATION 1: 0: 0 = 36001 TICKS

Source	Destination ---------->																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	NR Blocked by Source
1/CKS	0	0	0	4	0	4	0	3	1	3	0	0	0	0	1	2	0	0	3	0	1	0	19	41
2/FGB	1	0	0	3	0	5	1	13	0	0	0	1	0	0	0	1	2	0	0	0	0	0	1	28
3/FBK	3	1	0	13	0	46	4	3	5	2	0	0	0	0	1	0	0	0	0	0	0	0	14	92
4/YOK	11	2	9	0	2	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	49
5/CZA	1	0	13	4	0	20	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	14	53
6/YSU	14	4	30	2	3	0	2	1	1	1	0	0	0	0	0	2	4	1	2	0	2	0	22	91
7/1WK	3	0	4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
8/PRL	3	3	8	0	0	3	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	9	28
9/SCS	1	3	10	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	28
10/FSR	6	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	2	13
11/H1K	0	1	3	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	6	0	1	0	0	15
12/CSH	4	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	10
13/LOD	3	0	4	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	29
14/SLO	5	2	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	15
15/DRA	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
16/CON	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17/YSM	2	4	1	0	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	470	491
18/YSS	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	1	0	0	0	0	0	3	270	277
19/OSK	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	108	110
20/TGS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21/WLK	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	138	143
22/PSK	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	32
23/CRC	20	0	24	20	12	22	0	16	27	0	0	8	0	0	0	41	613	116	105	0	229	54	0	1307
Total Blocked To Destination	79	23	111	46	22	135	8	38	35	6	0	9	0	0	2	47	622	117	122	0	233	65	1142	= 2862

Figure A.4.

OFFERED CALL MATRIX FOR PRECEDENCE LEVEL 1
FROM 2: 0: 0 TO 3: 0: 0 DURATION 1: 0: 0 = 36001 TICKS

Source	Destination ----------> 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	NR Offered by Source
1/CKS	0	11	22	17	4	13	3	48	38	18	17	12	0	0	18	126	6	0	114	0	6	1	20	494
2/FGB	28	0	4	12	4	24	6	44	18	7	5	11	0	0	4	10	6	1	5	0	0	0	2	191
3/FBK	54	23	0	57	14	116	11	45	40	30	16	10	0	0	8	36	8	2	34	0	4	0	17	525
4/YOK	40	9	53	0	116	107	36	14	18	7	3	4	0	0	4	35	17	4	21	0	4	0	12	504
5/CZA	2	0	36	58	0	100	16	1	7	1	1	2	0	0	0	12	88	0	3	0	1	0	15	343
6/YSU	53	9	91	36	73	0	17	22	12	8	9	10	0	0	5	47	33	2	21	0	5	0	27	480
7/IWK	7	4	10	4	17	12	0	2	1	1	4	1	0	0	2	3	3	1	1	0	0	2	0	75
8/PRL	38	14	32	17	3	23	3	0	18	3	18	40	0	0	1	3	16	4	30	0	6	0	12	281
9/SCS	23	15	33	8	2	18	1	26	0	54	45	23	0	0	0	6	7	1	19	0	1	0	14	296
10/FSR	17	4	9	11	1	11	0	6	18	0	16	7	0	0	0	1	6	0	12	0	0	0	4	123
11/HIK	5	5	10	3	2	10	2	32	99	51	0	10	0	0	0	3	0	0	16	0	4	0	0	252
12/CSH	15	3	8	8	1	10	1	44	22	11	20	0	0	0	0	1	5	1	12	0	1	0	0	163
13/LOD	20	3	14	17	3	25	2	2	0	0	0	0	0	2	0	0	17	14	11	0	0	0	26	156
14/SLO	31	5	22	22	3	29	6	4	0	0	0	0	2	0	0	0	5	2	12	0	2	0	7	152
15/DRA	9	2	3	4	0	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	22
16/CON	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17/YSM	16	9	10	16	258	20	4	19	18	8	6	7	0	0	0	22	0	233	67	0	80	20	598	1411
18/YSS	7	3	5	9	6	3	2	10	5	5	2	3	0	0	0	14	343	0	94	0	58	11	346	926
19/OSK	76	1	29	7	3	2	2	16	15	12	4	8	0	0	0	29	47	13	0	0	65	11	140	480
20/TGS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21/WLK	6	0	3	6	3	10	1	4	5	1	0	1	0	0	0	0	89	82	63	0	0	26	172	472
22/PSK	7	0	2	3	1	0	0	0	0	1	2	2	0	0	0	1	10	6	5	0	23	0	40	103
23/CRC	21	0	29	27	13	27	0	23	30	0	2	8	0	0	0	56	772	154	128	0	284	66	0	1640
Total Offered To Destination	475	120	425	342	527	563	113	362	364	219	170	159	2	2	42	405	1478	520	668	0	544	137	1452	= 9089

Figure A.5.

CODE CANCELLED OR CODE GAPPED CALL MATRIX FOR PRECEDENCE LEVEL 1
FROM 2: 0: 0 TO 3: 0: 0 DURATION 1: 0: 0 = 36001 TICKS

Source	Destination ---------->																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	NR CANCELLED at Sour
1/CKS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2/FGB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3/FBK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
4/YOK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5/CZA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6/YSU	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
7/IWK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8/PRL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	2
9/SCS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10/FSR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11/HIK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12/CSH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13/LOD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14/SLO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15/DRA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16/CON	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17/YSM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44	0	0	0	44
18/YSS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	11
19/OSK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29	0	0	0	29
20/TGS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21/WLK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	25
22/PSK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	4
23/CRC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	26
Total Cancelled To Destination	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	143	0	0	0	143
	0		0		0		0		0		0		0		0		0		0		0		0	= 143

Figure A.6.

```
Oct 28 09:17 1987  hmh.sw.run4 Page 1

 Switch Report for Node  CRC
   At time   2: 0: 0   =   72000 TICKS
 Originating calls                          157     15      0      0      0
 Tandem calls                                 0      0      0      0      0
 Outgoing calls                             157     15      0      0      0
 Originating-outgoing calls NC              129      0      0      0      0
 Tandem calls NC                              0      0      0      0      0
 Calls code cancelled                         1      2      0      0      0
 Total calls to line busy                     0      0      0      0      0
 Incoming calls                              21      9      2      0      0
 Terminating calls                           21      9      2      0      0
CCAN       1       2       0       0       0
CALT       0       0       0       0       0
 Trunk Group Report for Land Link to YSM
 Access attempts                            156     13      0      0      0
 Overflows                                  129     13      0      0      0
 Incoming attempts                           21      9      2      0      0
 Usage (traffic)                            215
CANT
SKIP
 Trunk Group Report for Land Link to TGS
 Access attempts                            140      6      0      0      0
 Overflows                                  140      6      0      0      0
 Incoming attempts                            0      0      0      0      0
 Usage (traffic)                              0
CANT
SKIP


 Total calls in on these TGs =               21      9      2      0      0
 Total attempts to use links =              296     19      0      0      0
 Total times attempt failed =               269     19      0      0      0
 Nr. calls sent out over links  =            27      0      0      0      0
 Nr. calls routed out of switch =            27     13      0      0      0


 Switch Report for Node  YSM
   At time   2: 0: 0   =   72000 TICKS
 Originating calls                          131     24      6      0      0
 Tandem calls                                79     14      1      0      0
 Outgoing calls                             210     38      7      0      0
 Originating-outgoing calls NC               38      0      0      0      0
 Tandem calls NC                             43      0      0      0      0
 Calls code cancelled                         5      1      0      0      0
 Total calls to line busy                     0      0      0      0      0
 Incoming calls                             155     37      9      0      0
 Terminating calls                           76     23      8      0      0
CCAN       5       1       0       0       0
CALT       0       0       0       0       0
 Trunk Group Report for Land Link to YOK
 Access attempts                             19      6      1      0      0
 Overflows                                    5      3      1      0      0
 Incoming attempts                           13      1      0      0      0
 Usage (traffic)                             68
```

Figure A.7.

very poor. The switch report for YSM, which begins near the bottom of the page, is much longer because YSM has many more trunk groups than CRC. The length of the hmh.sw.run4 file is about 36 pages, and this was produced by only two switches in one hour; hence the desire to limit switch report files from the simulator to only time intervals of interest.

Figure A.8 is the first page of the compacted form hmh.exp.run4 of the switch report for this run, as it would be sent to the expert system. Labels are left off, and zeros are not printed. The total length of this file was 23 pages, each containing less than a third of the characters on the 36 pages of the hmh.sw.run4 report.

```
Oct 28 12:17 1987  hmh.exp.run4 Page 1

Node  CRC 72000
     157      15
       0
     157      15
     129
       0
       0
      21       9       2
      21       9       2
trunk   YSM L
     156      13
     129      13
      21       9       2
     215
trunk   TGS L
     140       6
     140       6
       0
       0
control-info
CCAN        1       2
END-NODE
Node  YSM 72000
     131      24       6
      79      14       1
     210      38       7
      38
      43
       0
     155      37       9
      76      23       8
trunk   YOK L
      19       6       1
       5       3       1
      13       1
      68
trunk   CZA L
      25       4       1
       8       1
       6       2       2
     102
trunk   PRL L
      10
       8
       2
      12
trunk   PRL S
       8
       4
       0
      15
trunk   SCS L
      12
      11
       5
      13
```

Figure A.8.

APPENDIX B

PROTOTYPE NETWORK MANAGEMENT EXPERT SYSTEM

The process of network management for the DSN has two major components, which are much the same for manual operation by skilled humans as for expert system software techniques. First, the management center must have a detailed and accurate description of the network, including static information such as topology and connectivity as well as dynamic information such as traffic patterns and current statistics. Second, the management center must have knowledge about how to optimize network performance. This knowledge must include techniques for recognizing incipient problem conditions on the basis of current information about the network, as well as for selecting and applying the right corrective actions from the available repertoire.

The complexity of this combination of requirements poses difficult problems for human network managers under any conditions, and this will be especially true when the DSN presents them with functions and control options that have not previously existed. It is highly appropriate to take advantage of two well-tested strong points of expert system techniques, namely effective internal representation of complex systems, and application of complex domain-specific knowledge to these representations. Accordingly, our goals are to use these techniques to develop suitable representations of DSN elements, and to apply all available knowledge -- standard algorithms, textbook procedures, expert wisdom, and the results of our experimental investigations -- to optimize network performance.

This Appendix describes our efforts in this area during the past year. It includes a description of the proposed architecture, a summary of current status, and an assessment of the potential of this methodology. It also presents an estimate of the magnitude and complexity of the tasks of fully implementing an expert system meeting the stated goals.

B-1. Coupling an Expert System with a Network Simulation

The specific context of our expert system development is the Call-by-Call Simulator (CCSIM), as modified and extended during the past year. During its simulation activity CCSIM generates the same series of 5-minute statistical reports that would be produced by each switch in the network, based on the switch's own local view of the traffic and events being simulated. This large volume of data will be presented to the expert system, which will identify problem areas and recommend control actions to be taken by CCSIM.

There are two major benefits in using a simulator while developing the expert system, namely rapid accumulation of empirical knowledge and convenient testing of system performance. The actual DSN will not be a source of network management knowledge until it is fully installed and carrying a full-scale traffic load. Even then, the network managers will have to wait until exceptional events occur naturally in order to learn about them, since it will be impossible to conduct deliberate performance-jeopardizing experiments with the operational DSN. Similarly, it will be impossible to run experiments in testing the ability of the expert system to select and apply the right control actions. With CCSIM it will be possible to explore many different variants of traffic patterns and

network damage scenarios to learn what to look for in switch reports and how to interpret the results. The identical simulations can be done as many times as desired in evaluating the selection and parameterization of control actions to overcome problems. The usefulness of this interactive process will continue after the DSN is fully installed, as well.

B-2. Knowledge Engineering

The environment described above, coupling a sophisticated network simulation with an expert system for network management, is ideal for knowledge engineering in this domain. By the term "knowledge engineering" we mean the process of accumulating and understanding the procedures, algorithms, and heuristics necessary to deliver expert performance in the domain, followed by encoding the knowledge and incorporating it into the expert system. Normally such knowledge must be gleaned from all available sources, including books, reports, and interviews with people in the target profession. The information possessed by these individuals is arguably the most important, and also the hardest to obtain. In the present effort we will also depend heavily upon analyzing a library of statistics summaries and switch reports that will be compiled as the output of many CCSIM runs, indicating what patterns evolve in the network with different combinations of traffic, damage, and control settings.

The knowledge engineering for this project is further complicated by the fact that it is not yet possible to validate CCSIM by comparing it with an existing system. For the time being we will undertake to establish confidence in CCSIM by calibrating it against other existing tools in use for analyzing the DSN.

INTEGRATED SYSTEM ARCHITECTURE

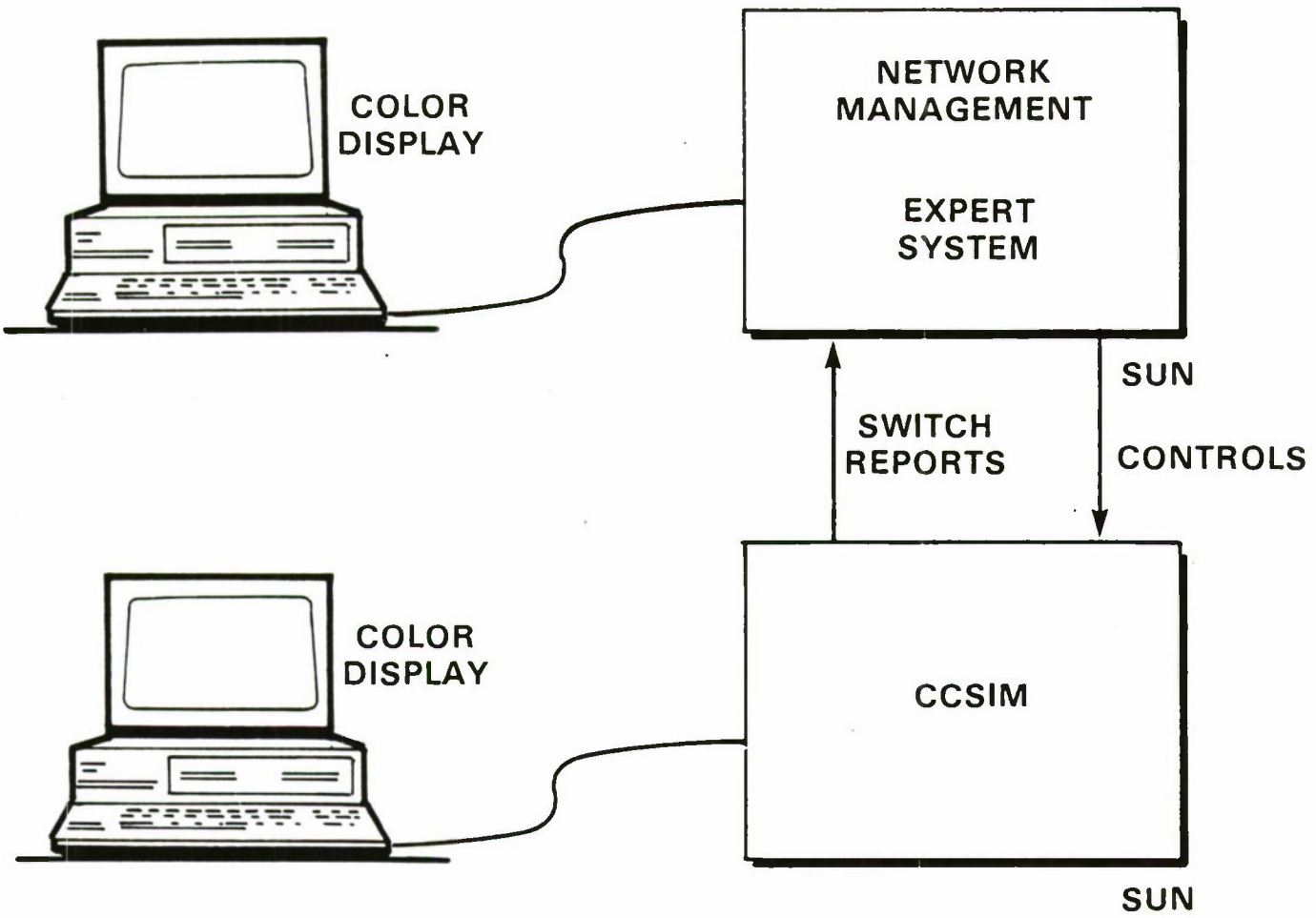


Figure B.1.

B-3. Architecture of Prototype Expert System

The following sections describe the initial design of our proposed network management expert system, as well as the current status of our implementation.

B-3.1 General Hardware and Software Configuration

Figure B.1 is a top-level hardware configuration diagram of the target system. The system consists of two identical SUN work stations with interactive color monitors, linked by an Ethernet. The hardware and software of the work station running CCSIM have been described in previous sections. To create a suitable expert system environment, two major additions were made to the UNIX operating system environment of the second work station, namely SUN Common LISP and ART. The former is a full implementation of the currently proposed Common LISP standard, with a number of additional features. It was written by Lucid, Inc., and is marketed by SUN. ART (for Automated Reasoning Tool) is an expert system building tool marketed by the Inference Corporation. It provides a ready-made shell containing many of the domain-independent mechanisms common to typical expert systems. Its critical features with respect to this project include a frame-based data representation, forward- and backward-chaining rule formats and inference mechanisms, and a powerful pattern matching facility.

Since both LISP and ART require large memory sizes, we have added an 8-megabyte memory board to the expert system machine, bringing its total memory size to 16 megabytes. We have also reconfigured the machine with 80 megabytes of swap space for disk interactions. Graphics representation

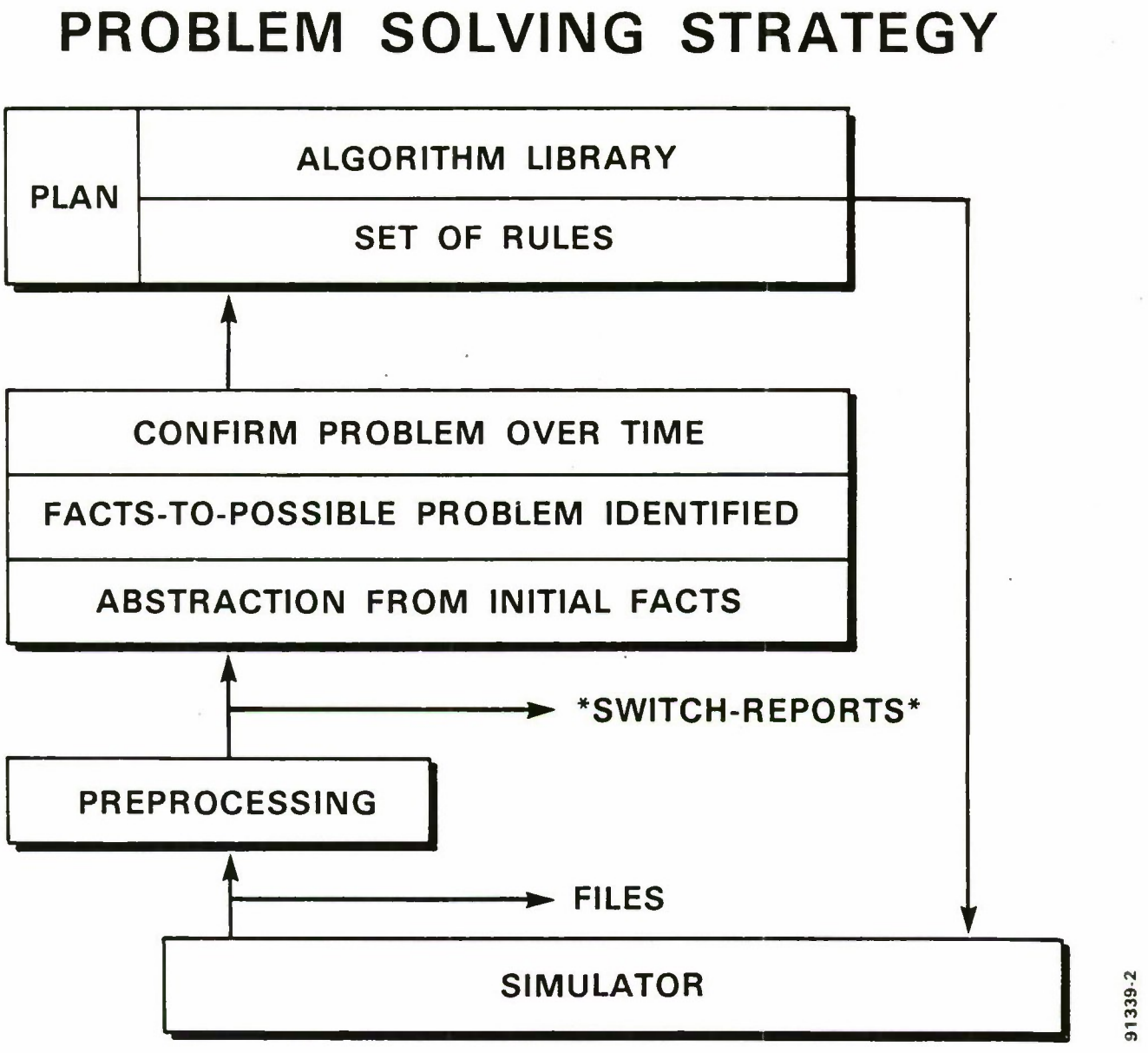

PROBLEM SOLVING STRATEGY
PLAN
ALGORITHM LIBRARY
SET OF RULES
CONFIRM PROBLEM OVER TIME
FACTS-TO-POSSIBLE PROBLEM IDENTIFIED
ABSTRACTION FROM INITIAL FACTS
SWITCH-REPORTS
PREPROCESSING
FILES
SIMULATOR
91339-2

Figure B.2.

issues have posed certain problems: although both ART and SUN Common LISP provide windowing and graphics capabilities, our past experience with ART and a study of the facilities within SUN Common LISP convinced us that neither was appropriate for the current project. Consequently we have chosen to use SUNview, a sophisticated C-based graphics package available from SUN that can be accessed from LISP programs by means of a set of LISP macros, also available from SUN. As windowing and graphics packages from SUN and the LISP vendors evolve, it is quite possible that we will choose to convert to a more homogeneous LISP-based combination. One possible candidate is the Symbolic Programming Environment (SPE) recently announced by SUN, that will be available in the spring of 1988.

B-3.2 Details of Software Architecture

Figure B.2 shows a top-level diagram of our visualization of interactions among the major software modules of the expert system. At the lower left, switch reports are received in highly abbreviated format from the simulator. The preprocessing module expands these reports into a symbolic format which is comprehensible to ART. The three boxes in the center together constitute a problem recognition module, which acts upon the switch reports to identify and confirm problems in the network. Descriptions of these problems are then passed to the plan generation module at the top of the figure, which selects a set of switch control actions that will optimize network performance in the presence of the problem.

The following paragraphs describe the functions of preprocessing, problem recognition, and plan generation. First, in order to better

01	CKS	CKS	PHILIPPI			15.13-120.55	Y	MFS	CLARK	RP	JUL87
02	FGB	FGB	GUAM			13.58-144.92	Y	SAS	FINEGAYNGQ		OCT85
03	FBK	FBK	JAPAN-OK			26.28-127.76	Y	SAS	FTBUCKNRJA		OCT88
04	YOK	YOK	JAPAN-ML	GW	0.05	35.73-139.34	Y	SAS	YOKOTA	JA	OCT85
05	CZA	CZA	JAPAN-ML	GW	0.05	35.48-139.39	Y	MFS	CP ZAMA	JA	OCT88
06	YSU	YSU	JAPAN-ML		0.05	35.40-139.62	Y	MFS	YOKOSUKA	JA	OCT88
07	IWK	IWK	JAPAN-ML		0.05	34.07-132.23	N	MFS	IWAKUNI	JA	OCT88
08	PRL	PRL	HAWAII	GW	0.01	21.35 157.95	Y	MFS	PERLHRBR15		AUG87
09	SCS	SCS	HAWAII	GW	0.01	21.48 158.08	Y	MFS	SCHFLDBK15		FEB88
10	FSR	FSR	HAWAII		0.01	21.34 157.87	L	MFS	FTSHAFTR15		DEC87
11	HIK	HIK	HAWAII		0.01	21.34 157.96	L	MFS	HICKAM	15	DEC87
12	CSH	CSH	HAWAII		0.01	21.37 157.92	L	MFS	CP SMITH15		FEB88
13	LOD	LOD	CONUS	GW		38.18 121.26	Y	SAS	LODI	06	
14	SLO	SLO	CONUS	GW		35.27 120.61	Y	SAS	SNLSOBSP06		
15	DRA	DRA	CONUS			39.01 77.33	Y	SAS	DRANSVLL51		
16	CON	CON	CONUS			39.01 76.94	Y	SAS	FICTICIOUS		
17	YSM	YSM	KOREA		0.05	37.51-126.85	L	MFS	YONGSNMNKS		APR86
18	YSS	YSS	KOREA		0.05	37.50-126.85	L	MFS	YONGSNSOKS		APR86
19	OSK	OSK	KOREA		0.05	37.10-127.03	L	MFS	OSAN	KS	JUL87
20	TGS	TGS	KOREA		0.05	37.34-127.01	L	MFS	TANGO	KS	OCT86
21	WLK	WLK	KOREA		0.05	35.86-128.70	L	MFS	CPWALKERKS		JAN87
22	PSK	PSK	KOREA		0.05	35.12-129.06	N	MFS	PUSAN	KS	OCT87
23	CRC	CRC	KOREA		0.05	37.75-127.02	N	MFS	CPREDCLDKS		JAN88

Figure B.3.

understand the knowledge and data representation environment of the expert system in general, we will describe the network database configuration.

B-3.3 Database Representation of Network Topology and Switch Reports

In order for the expert system to take actions that correspond correctly to the topology of the network, the topology must be represented in a format which the expert system can "understand". The topology information as received from the DCA for the 23-node Pacific DSN backbone is shown in Fig. B.3 (the node information) and Fig. B.4 (the trunk connectivity data).

Because the DSN design is continuing to evolve, it is likely that we will often want to replace the node and trunk information in the expert system with new data. Since manual conversion of the raw data to ART's internal representation formats is a tedious and time-consuming task, we decided that it would be wise to write software that would do the conversion process algorithmically. At the same time, we wrote the algorithms necessary to allow the user to interactively build the graphical display of the network on the SUN color monitor. Finally, algorithms were written to convert the display coordinates from the representation used by the expert system to that required by CCSIM. This assures that the network graphics displays will be identical for CCSIM and the expert system. The result of all this is that the expert system can be reconfigured in a matter of minutes, both internally and graphically, when new database files are received from the DCA.

The exact format of the internal network representation is very important, because the ART rules look for precise pattern matches in this

```
 0  7 14 20  0  0  0 12 12  0  0  0  6  6  0  0  0  0  9  0  0  0  0
 3  0 13  7  0  0  0  9  9  0  0  0  0  0  0  0  0  0  0  0  0  0  0
14 10  0  6  5  0  3 10  9  0  0  0  0  0  0  0  0  0  0  6  0  5  0  0
03  3 18  0  0  0  0 16 17  0  0  0 18 18  0  0  0  0  0  0  0  0  0
 0  0 17 48  0  0  0 16 17  0  0  0  8  8  0  0  0  0  0  0  0  0  0
 0  0  0 48 70  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  9 17 22  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 6  3  5  5  5  0  0  0  0  0  0  0 33 18 48  0  6  0  3  0  4  0  0
 6  3  5  5  5  0 0259  0  0  0  0 18 37 18  0  6  0  3  0  4  0  0
 0  0  0  0  0  0 0109204  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0 0153112 47  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0 0170 94 93 63  0  0  0  0  0  0  0  0  0  0  0  0
 2  0  0  3  2  0  0 15  6  0  0  0  0  0  0  0  4  0  0  2  2  0  0
 1  0  0  2  1  0  0  6 11  0  0  0 43  0  0  0  4  0  0  2  2  0  0
 0  0  0  0  0  0  0  0  0  0  0 0299299  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0 0299299299  0  0  0  0  0  0  0  0
 0  0  0 24 36  0  0  5  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 0192  0  0  0  0  0  0
 8  0  6 22  0  0  0  4  4  0  0  0  0  0  0  0 48 59  0  0  0  0  0
 0  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0 72 48 15  0  0  0  0
 0  0  4  0  4  0  0  4  4  0  0  0  0  0  0  0 72 72 37 13  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 15  0 20  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 72  0  0 48  0  0  0
```

Figure B.4.

representation. Figures B.5 and B.6 show sample schemata for a node and a trunk ("schema", plural "schemata", is ART's term for their frame representation for object descriptions). These objects are organized in a many-layered hierarchical tree by ART, with objects at a lower level being "instances-of" more general objects at higher levels. A major advantage of this organization is that objects "inherit" information and characteristics from the higher objects of which they are instances; for example, node CKS inherits the entire list of properties that have been typed in only once for the generic object named "node". Besides providing important reductions in typing labor, this structure is helpful in organizing the huge databases needed for DSN network management.

B-3.4 Preprocessing

Besides the network structure, the other major information input to the expert system is a continuous flow of status and statistics reports from the switches in the simulated network in CCSIM. Figure A.7 was an example of a switch report in expanded format, labeled and identified for programmer convenience, while Fig. A.8 showed the same report in the abbreviated format used to conserve bandwidth on the link between CCSIM and the expert system. The preprocessing module in the expert system converts the abbreviated switch reports into the specific ART representations required for subsequent operations. Figure B.7 shows a switch report after preprocessing, in a schema format similar to that of the node and trunk objects described above. The big difference is that the node and trunk definitions are constant throughout the current execution of the expert system, while the switch reports are continually changing. In ART

TRUNK SCHEMA

```
(DEFSCHEMA CKS-FBK-L
        (INSTANCE-OF      LAND-LINK)
        (END-NODES        (CKS FBK))
        (CAPACITY         14)
        (DIRECTIONALIZATION)
                  ●
                  ●
        )
```

Figure B.5.

NODE SCHEMA

```
(DEFSCHEMA CKS
        (INSTANCE-OF NODE)
        (OPERATED-BY)
        (NODE-LOCATION PHILIPPI)
        (HAS-CONNECTIVITY-WITH)
        (HAS-DH-CONNECTIVITY-WITH)
                    ●
                    ●
        )
```

Figure B.6.

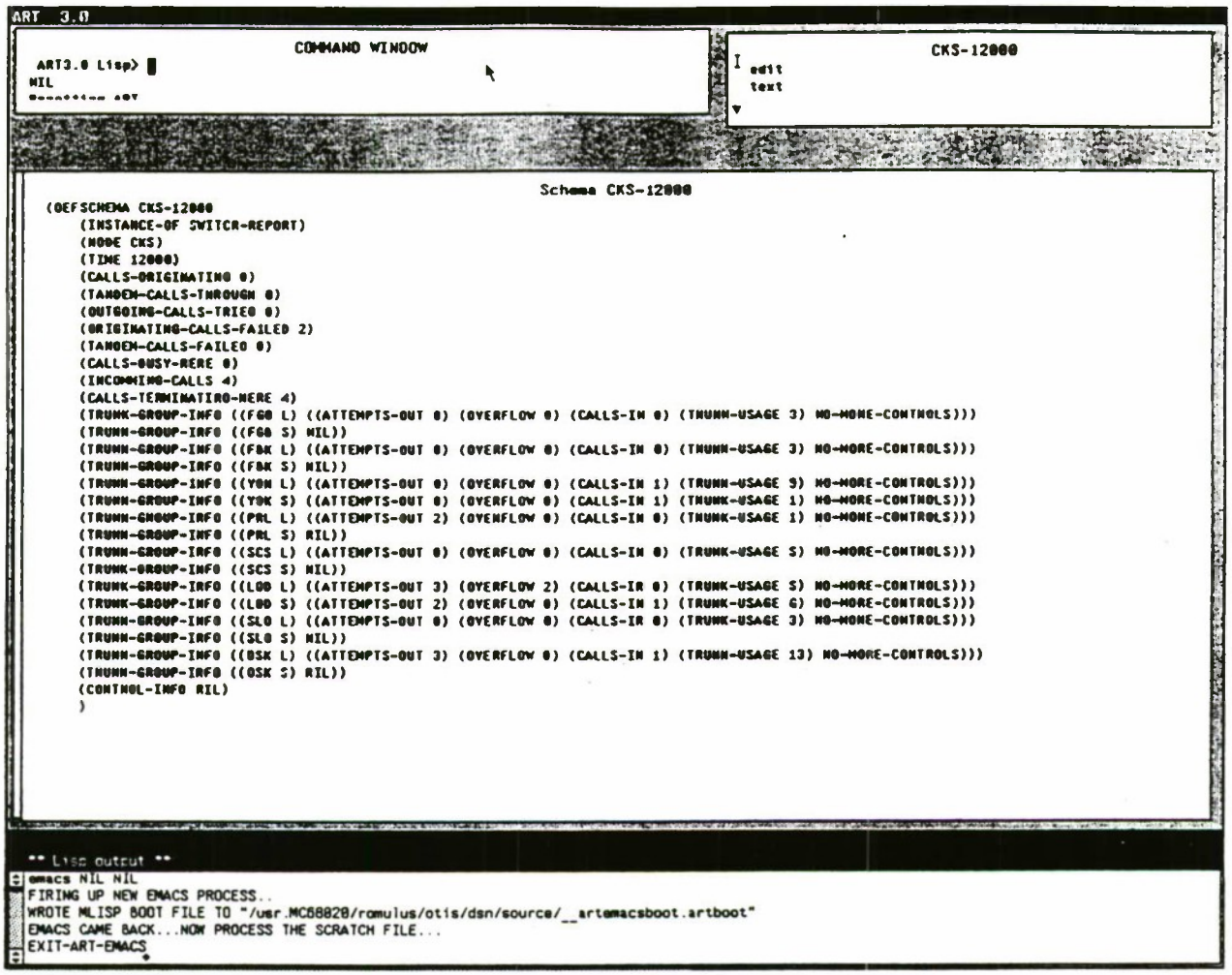

ART 3.0
COMMAND WINDOW
ART3.0 Lisp>
NIL
CKS-12000
edit
text
Schema CKS-12000
(DEFSCHEMA CKS-12000
(INSTANCE-OF SWITCH-REPORT)
(NODE CKS)
(TIME 12000)
(CALLS-ORIGINATING 0)
(TANDEM-CALLS-THROUGH 0)
(OUTGOING-CALLS-TRIED 0)
(ORIGINATING-CALLS-FAILED 2)
(TANDEM-CALLS-FAILED 0)
(CALLS-BUSY-HERE 0)
(INCOMMING-CALLS 4)
(CALLS-TERMINATING-HERE 4)
(TRUNK-GROUP-INFO ((FGD L) ((ATTEMPTS-OUT 0) (OVERFLOW 0) (CALLS-IN 0) (TRUNK-USAGE 3) NO-MORE-CONTROLS)))
(TRUNK-GROUP-INFO ((FGD S) NIL))
(TRUNK-GROUP-INFO ((FBK L) ((ATTEMPTS-OUT 0) (OVERFLOW 0) (CALLS-IN 0) (TRUNK-USAGE 3) NO-MORE-CONTROLS)))
(TRUNK-GROUP-INFO ((FBK S) NIL))
(TRUNK-GROUP-INFO ((YBH L) ((ATTEMPTS-OUT 0) (OVERFLOW 0) (CALLS-IN 1) (TRUNK-USAGE 9) NO-MORE-CONTROLS)))
(TRUNK-GROUP-INFO ((YBK S) ((ATTEMPTS-OUT 0) (OVERFLOW 0) (CALLS-IN 1) (TRUNK-USAGE 1) NO-MORE-CONTROLS)))
(TRUNK-GROUP-INFO ((PRL L) ((ATTEMPTS-OUT 2) (OVERFLOW 0) (CALLS-IN 0) (TRUNK-USAGE 1) NO-MORE-CONTROLS)))
(TRUNK-GROUP-INFO ((PRL S) RIL))
(TRUNK-GROUP-INFO ((SCS L) ((ATTEMPTS-OUT 0) (OVERFLOW 0) (CALLS-IN 0) (TRUNK-USAGE S) NO-MORE-CONTROLS)))
(TRUNK-GROUP-INFO ((SCS S) NIL))
(TRUNK-GROUP-INFO ((LOD L) ((ATTEMPTS-OUT 3) (OVERFLOW 2) (CALLS-IN 0) (TRUNK-USAGE S) NO-MORE-CONTROLS)))
(TRUNK-GROUP-INFO ((LOD S) ((ATTEMPTS-OUT 2) (OVERFLOW 0) (CALLS-IN 1) (TRUNK-USAGE 6) NO-MORE-CONTROLS)))
(TRUNK-GROUP-INFO ((SLO L) ((ATTEMPTS-OUT 0) (OVERFLOW 0) (CALLS-IN 0) (TRUNK-USAGE 3) NO-MORE-CONTROLS)))
(TRUNK-GROUP-INFO ((SLO S) NIL))
(TRUNK-GROUP-INFO ((OSK L) ((ATTEMPTS-OUT 3) (OVERFLOW 0) (CALLS-IN 1) (TRUNK-USAGE 13) NO-MORE-CONTROLS)))
(TRUNK-GROUP-INFO ((OSK S) RIL))
(CONTROL-INFO RIL)
)
** Lisp output **
emacs NIL NIL
FIRING UP NEW EMACS PROCESS...
WROTE MLISP BOOT FILE TO "/usr.MC68020/romulus/otis/dsn/source/__artemacsboot.artboot"
EMACS CAME BACK...NOW PROCESS THE SCRATCH FILE...
EXIT-ART-EMACS

Figure B.7.

terminology, the preprocessed switch report schemata are continually being "asserted" and "retracted" as facts in the expert system database. Since the ART inference engine is continually scanning the database for fact patterns that match rules, new switch reports are acted upon by rules as soon as they are asserted.

The details have been worked out for the communication mechanisms among processes and between work stations that will move switch reports from CCSIM to the expert system, and will return control information from the expert system to CCSIM. These mechanisms have been successfully tested, but are not yet being used in an on-line sense. For the time being CCSIM also dumps switch reports into files which the expert system can subsequently retrieve and process. This procedure facilitates software development by allowing work on CCSIM and the expert system to proceed independently of each other. It also conveniently supports repeated expert system runs with the identical set of switch reports, as an aid in developing and debugging rules.

One additional function has already been implemented in the preprocessing module, and no doubt others will be added in the future. This extra function is to "remember" past reports from each switch node, in the form of lists of LISP flavor instances, after they have been retracted from the active ART database. These lists can vary in length as determined by the expert system, depending on how much past history it feels it should be evaluating.

B-3.5 Problem Recognition and Control Selection

Four fundamental steps are required for the expert system to function as desired:

EXPERT SYSTEM STRUCTURE

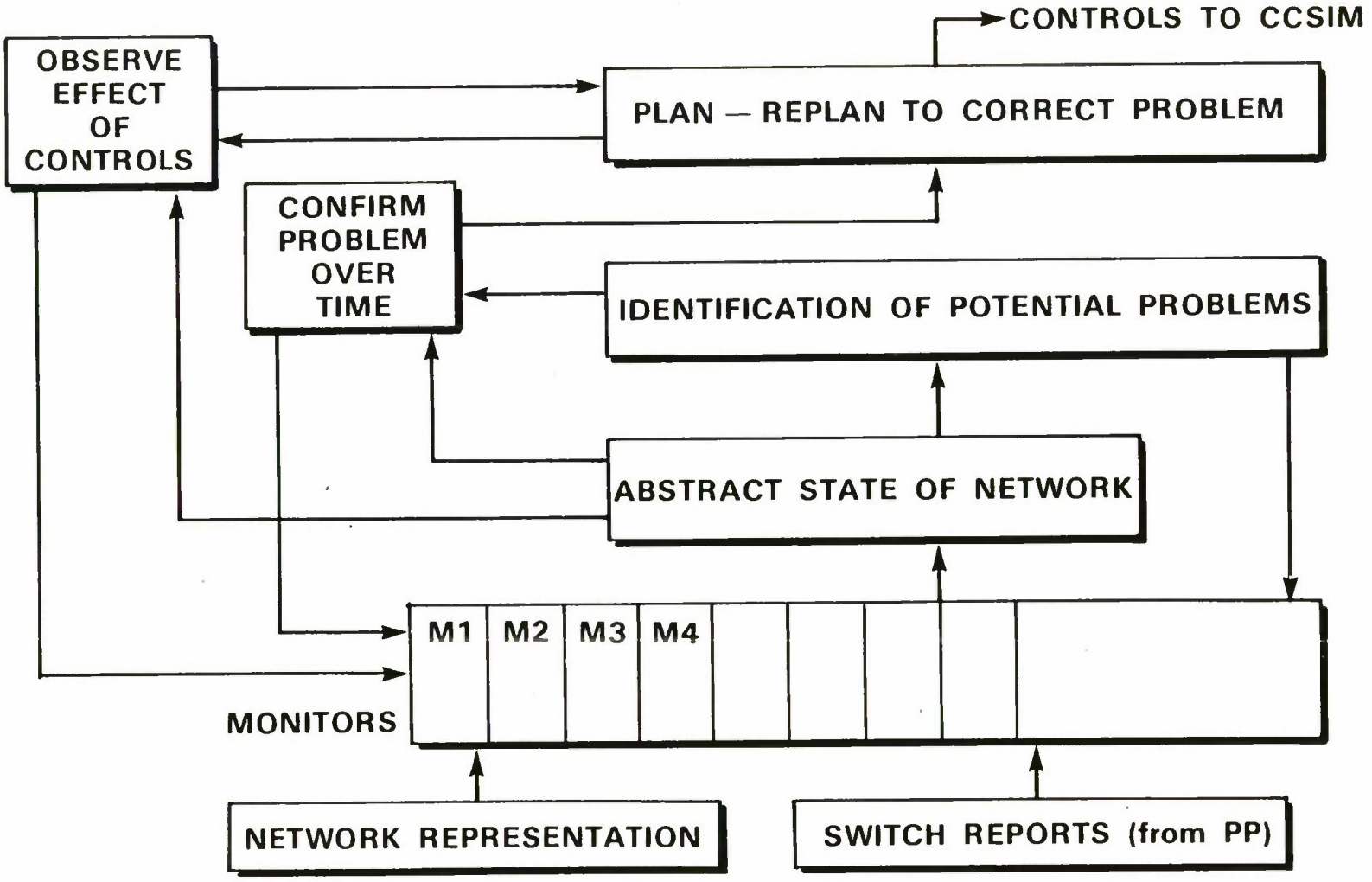


Figure B.8.

(1) Understand the patterns in switch report data that correspond to network problem conditions, and especially those patterns which foretell problems before they become severe;

(2) Implement expert system software that recognizes these patterns;

(3) Understand the selection and application of network controls to correct the identified problem conditions; and

(4) Implement expert system software that can do the selection and application of the controls.

All four of these steps inherently include the reverse process, that of altering and finally removing controls as the problems become corrected (by subsidence of congestion conditions, by repair of damaged facilities, or whatever else is appropriate). In particular, all the available controls function by diverting, limiting or blocking some kind of network facilities; therefore a control that is still in place after network repair represents a degradation in achievable service, which should be recognized and corrected.

Steps (1) and (3) above are the main business of the knowledge engineering which has gone on to some degree this year with the upgraded CCSIM, and will be carried out extensively in the coming year, as described elsewhere in this report. The topics of this Appendix are (2) and (4), and can be treated at length without yet knowing all the knowledge engineering results. Much can be implemented and demonstrated initially with classes of network problems for which the switch reports patterns are clearly definable, such as missing reports from damaged nodes.

The problem-solving strategy of Fig. B.2 is reflected in the proposed expert system architecture illustrated in Fig. B.8. The system inputs,

shown at the bottom, are the network representation and the preprocessed switch reports described earlier. A set of "monitor" modules continually analyze these inputs. Each monitor is assigned the task of recognizing one particular member of the repertoire of interesting patterns. The modularity of this design provides for orderly introduction of successive new additions to the catalog of patterns. A monitor may consist of one or more ART rules, or a combination of ART rules and Common LISP procedures.

Initial implementation of these monitors has begun, and three examples will be presented here. Three points should be noted, illustrating the potential power of this design approach, while reviewing these examples:

(1) Flexible pattern matching: a monitor that can recognize one instance of a pattern can be expected to recognize similar patterns anywhere in the network.

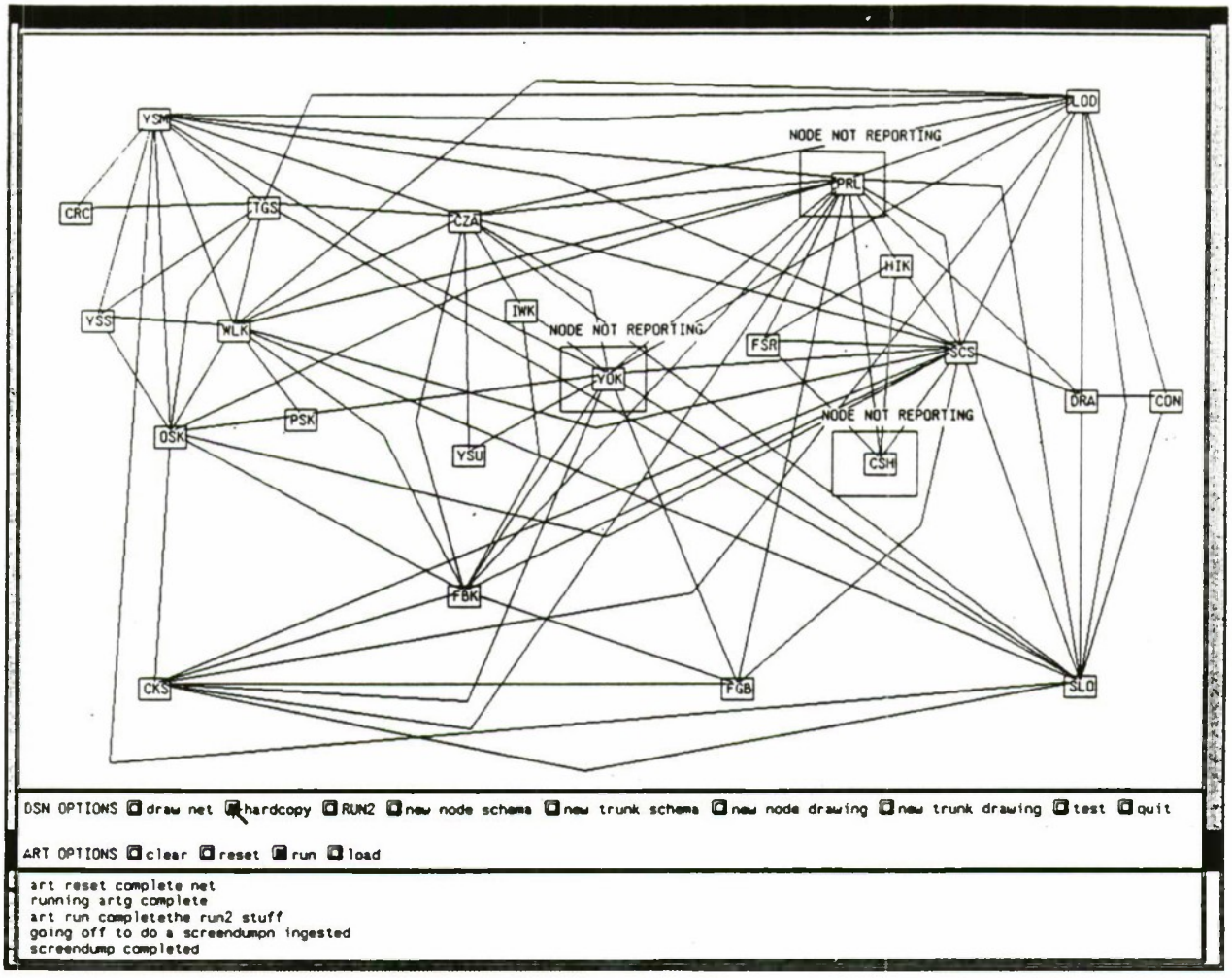
(2) Highly modular programming approach: as knowledge engineering proceeds, the types and characteristics of patterns of interest will continually evolve. Because the monitors are essentially independent of each other, anyone with a basic understanding of the rule format and of Common LISP can sit down and experiment to isolate patterns of any type.

(3) Easy activation/deactivation of monitors by assertions of facts: higher logical levels of the system can determine how carefully the inputs must be scrutinized. When the system observes that network operation is going smoothly, for example, it may shut off the bombardment of details from lower-level monitors. When first indications of a problem are observed, the higher logic can choose a tailored set of specific monitors to get a detailed look at factors relevant to the suspected type of problem.

Figures B-9, B-10, and B-11 show examples of one-rule monitors. The upper section of each figure is the graphics display produced by the expert system after applying the rule in the lower section, for a particular set of switch reports. The rule in Fig. B.9 locates all nodes that have failed to send in a switch report in the current interval; this would be used as contributory evidence in trying to decide whether such nodes had gone down. The ART syntax of the rule carries out the following search, expressed in English: "Locate every node-instance for which you cannot find a switch-report-instance having the name of that node-instance in its node-name slot. Every time you find such a node-instance, put a box around its graphic and label it NODE NOT REPORTING."

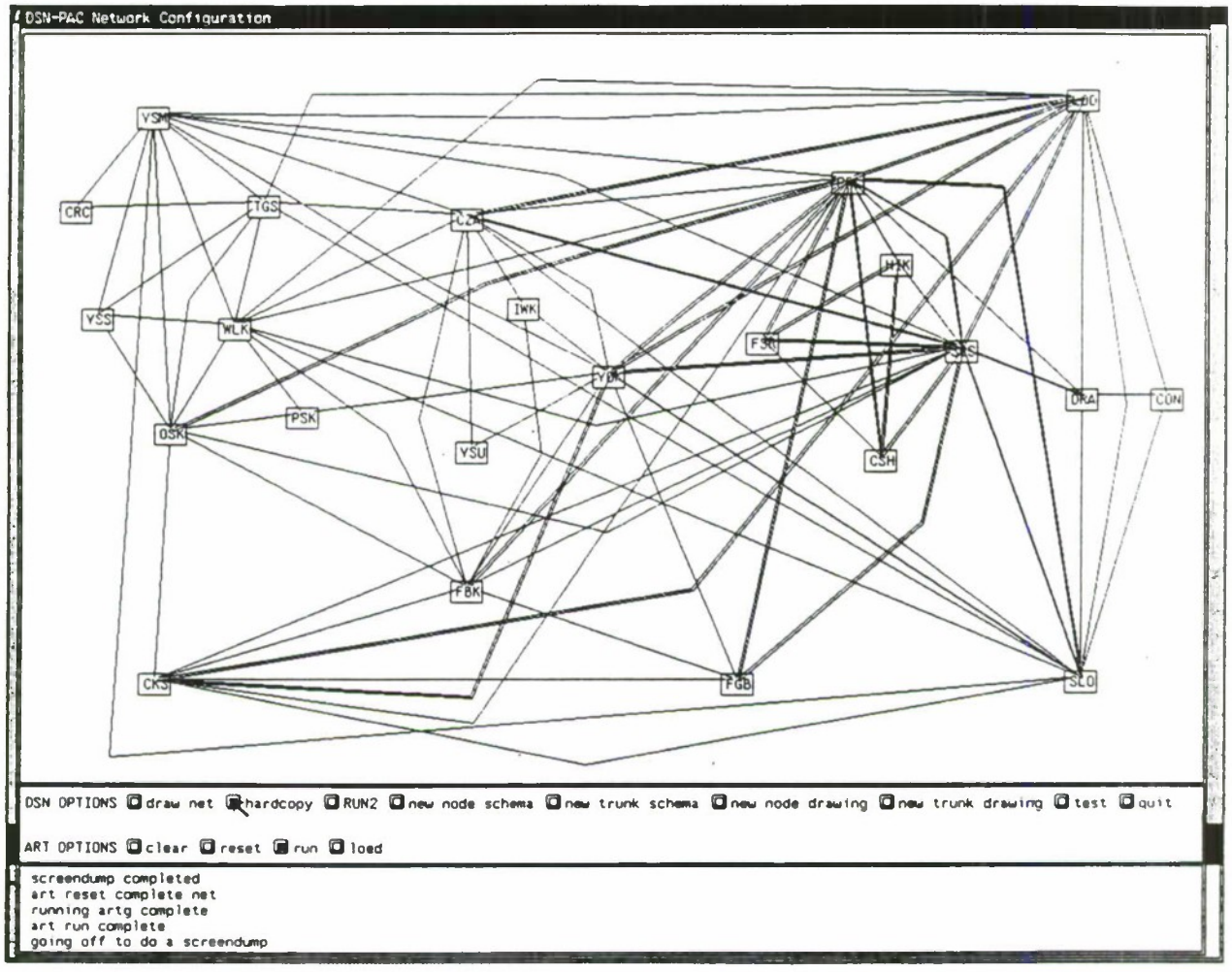
The one-rule monitor in Fig. B.10 locates trunks on which overflows (i.e., unsuccessful routing attempts) are currently being reported, and marks all such trunks with double line width. If many of the trunk groups to a particular node were experiencing overflows, this would be additional evidence of a possible outage of the node. The monitor in Fig. B.11 is somewhat more complicated; it locates every node whose neighbors are all reporting that they are receiving no incoming calls from the node.

In addition to altering the graphic display maintained by the expert system, outputs from the monitors will contribute to the evolving abstract representation of the network state, illustrated in the right center of Fig. B.8. This abstract representation will contain only items of current interest in the network management problem. The expert system database (including the complete network representation as well as all the current switch reports) contains a very large amount of information, only a small



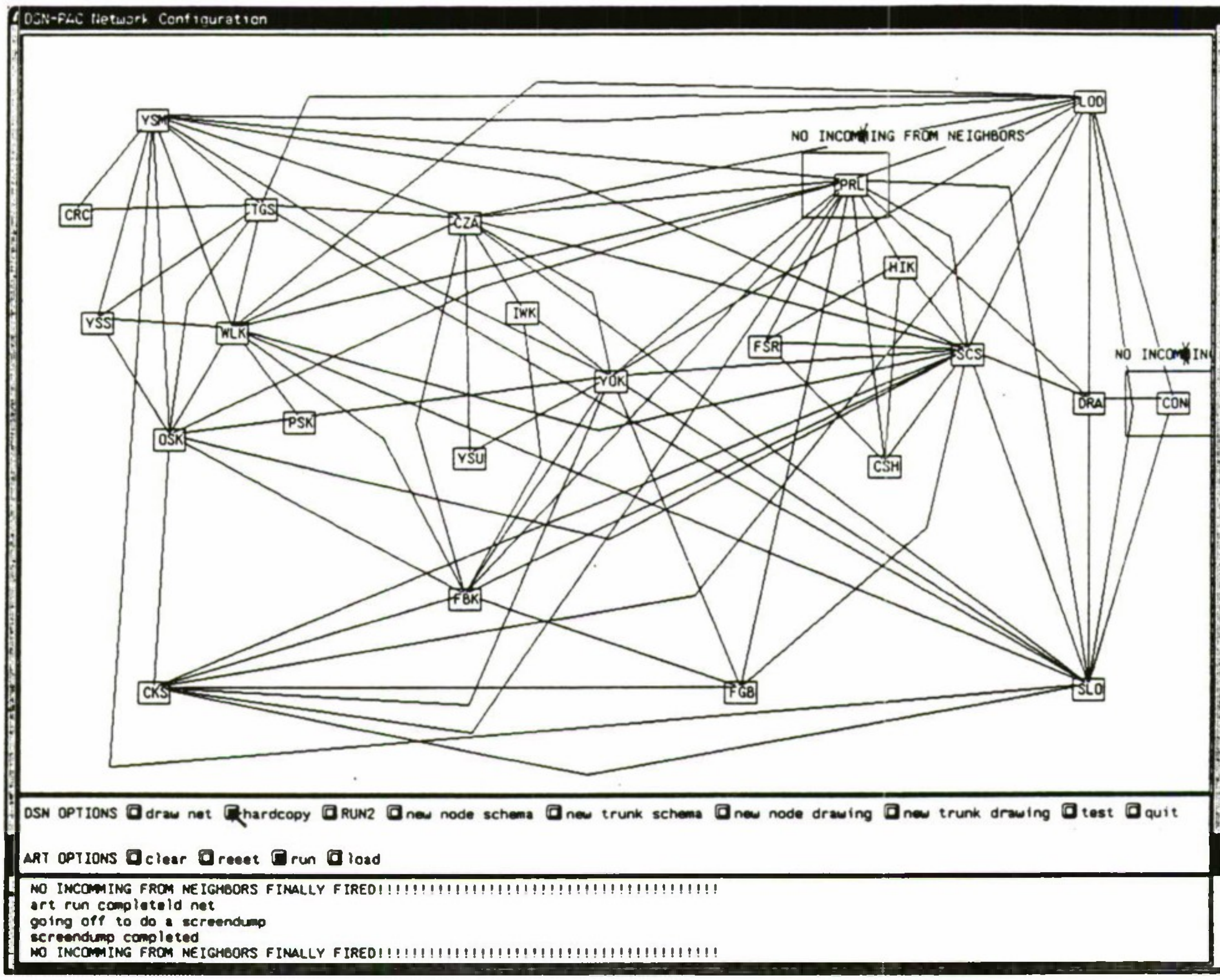

```
(defrule locate-node-not-responding
  (schema ?node-name
          (instance-of node))
  (not (schema ?switch-report
               (instance-of switch-report)
               (node ?node-name)))
=>
  (mark-node ?node-name "NODE NOT REPORTING")
```

Figure B.9.




```
(defrule locate-overflowed-trunks
  (schema ?switch-report
          (node ?node-name)
          (instance-of switch-report)
          (trunk-group-info ((?distant-end ?s-or-l)
                             ((ATTEMPTS-OUT ?) (OVERFLOW ~0)
                              (CALLS-IN ?) (TRUNK-USAGE ?)
                              NO-MORE-CONTROLS))))
=>
  (mark-overflow-on-trunk ?node-name ?distant-end ?s-or-l)
```

Figure B.10.




```
(defrule locate-neighbors-see-no-incoming-signals
  (schema ?node
          (instance-of node))
  (neighbor-alist (?node ?neighbor-alist))
  (for ?neighbor in$ ?neighbor-alist do
       (schema ?switch-report
               (instance-of switch-report)
               (node ?neighbor)
               (or (trunk-group-info ((?node ?s-or-l)
                                      ((ATTEMPTS-OUT ?) (OVERFLOW ?)
                                       (CALLS-IN 0) (TRUNK-USAGE ?)
                                       NO-MORE-CONTROLS)))
                   (trunk-group-info ((?node ?s-or-l) nil)))))

=>
  (mark-node ?node "NO INCOMING FROM NEIGHBORS")
```

Figure B.11.

part of which is relevant at any given time. In this sense the monitors act as filters to select only the database information that is of current interest to the upper levels of the expert system. The business of the knowledge engineering process is to provide guidance as to which items are "of current interest".

The abstract state of the network will be the input for three upper-level system modules in Fig. B.8. The "Identification of Potential Problems" module will scan the Abstract State for early indications of incipient trouble. It will report its suspicions to the "Confirm Problem Over Time" module, to see whether the problem persists or changes. Both the identification and confirmation modules can make suitable changes in the array of monitors that are turned on.

Once a problem has been confirmed over time, the "Plan - Replan" module is invoked to generate a set of control actions to correct the network performance. These controls are sent to CCSIM, and subsequent switch reports from CCSIM should show their effects. Since controls will not always produce exactly the results desired, the Planner must invoke the "Observe Effect" module, which can select its own set of monitors to see whether the control actions were successful. If they were not, the Observer reports the discrepancies to the Planner, who must then generate a modified set of control actions.

The expert system must also be able to determine when controls should be removed. The Observer must look for signs that a problem is going away. This can be very difficult, because the controls applied to combat the problem may hide the very indications the Observer is looking for. One

strategy that may be necessary is to remove controls gradually, or to occasionally remove controls in force for a brief interval, watch the results, and decide whether to reinstate them.

B-4. Prototype Expert System Status

The system design described in the previous section is currently being implemented. A flexible environment has been created which combines ART, SUN Common LISP, UNIX, and access to the SUNview graphics facilities, and it appears to have the necessary robustness to support future development. The representations for the nodes, trunks, switch reports and preprocessing module have also been created. The upper-level architecture of the system has been designed, and a few monitors have been implemented.

CCSIM is ready to support a concentrated program of knowledge engineering, as described above, and the way is now open to design and implement the succession of expert system rules and monitors that will carry out the management of the simulated network.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) K-BSACDSNTA	5. MONITORING ORGANIZATION REPORT NUMBER(S) ESD-TR-87-268
6a. NAME OF PERFORMING ORGANIZATION Lincoln Laboratory, MIT 6b. OFFICE SYMBOL *(If applicable)*	7a. NAME OF MONITORING ORGANIZATION Electronic Systems Division
6c. ADDRESS *(City, State, and Zip Code)* P.O. Box 73 Lexington, MA 02173-0073	7b. ADDRESS *(City, State, and Zip Code)* Hanscom AFB, MA 01731
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Air Force Systems Command, USAF 8b. OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-85-C-0002 Program 295
8c. ADDRESS *(City, State, and Zip Code)* Andrews AFB Washington, DC 20334	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 62702F; PROJECT NO.; TASK NO.; WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*

Knowledge-Based System Analysis and Control Defense Switched Network Task Areas

12. PERSONAL AUTHOR(S)
Harold M. Heggestad

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
Annual Report	FROM 1 Oct 86 TO 30 Sep 87	30 September 1987	116

16. SUPPLEMENTARY NOTATION

None

17. COSATI CODES			18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUB-GROUP	technical control; network management; expert systems; knowledge engineering; machine intelligence

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Extensive modifications have been made in the existing Defense Switched Network (DSN) Call-by-Call Simulator (CCSIM) to permit its use in learning how to apply Network Management techniques in the DSN. The initial capabilities and purposes of the CCSIM are described, and the requirements for the modifications are explained. In particular, the current DSN Network Management control options are defined and explained. The upgrading of CCSIM has been completed, and some hundreds of simulation runs have been carried out and analyzed to date. In support of this analysis, it has been necessary to augment the traditional Grade of Service (GOS) measure of network performance with a pair of new measures expressing actual user experience with the net, namely Call Failure Rate (CFR) and Mean Tries for Success (MTFS). A set of basic recommendations have been derived for near-term manual DSN Network Management actions under certain failure scenarios. Concurrently, a prototype has been developed for a knowledge-based Expert System to incorporate and organize the information and techniques being gathered, so that this knowledge can be made available for reference and advice for DSN Network Managers in the field.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Lt. Col. Hugh L. Southall, USAF	22b. TELEPHONE *(Include Area Code)* (617) 981-2330 22c. OFFICE SYMBOL ESD/TML